NASA Conference Publication 3025

Proceedings of the Scientific Data Compression Workshop

H. K. Ramapriyan, *Editor*
NASA Goddard Space Flight Center
Greenbelt, Maryland

Proceedings of a workshop sponsored by the
National Aeronautics and Space Administration,
Washington, D.C., and held at
Snowbird Conference Center
Snowbird, Utah
May 3-5, 1988

National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1989

TABLE OF CONTENTS

SUBPANEL RESULTS AND RECOMMENDATIONS

APPLICATION REQUIREMENTS AND CONSTRAINTS

Session Coordinator: Dave Nichols
Jet Propulsion Laboratory

PRECEDING PAGE BLANK NOT FILMED

COMMUNICATIONS REQUIREMENTS AND CONSTRAINTS

Session Coordinator: Edward C. Posner
Jet Propulsion Laboratory

COMPRESSION METHODS OVERVIEW

Session Coordinator: Tom Stockham
University of Utah

DATA COMPRESSION CASE STUDIES

Session Coordinator: Hampapuram Ramapriyan
Goddard Space Flight Center

DATA COMPRESSION ALGORITHM R&D

Session Coordinator: James A. Storer
Brandeis University

DATA COMPRESSION HARDWARE

Session Coordinator: Dave Eisenman
Jet Propulsion Laboratory

FOREWORD

Continuing advances in space and Earth science knowledge require increasing amounts of data to be gathered from spaceborne sensors. NASA expects to launch sensors during the next two decades which will be capable of producing an aggregate of 1500 Megabits per second if operated simultaneously. Two examples of high-rate sensors are the High Resolution Imaging Spectrometer (HIRIS) and the Synthetic Aperture Radar (SAR) to be flown on the Earth Observing System's (EOS) polar-orbiting platforms. These instruments are each capable of producing over 300 megabits/sec. There are several materials science experiments being designed for the Space Station whose aggregate bandwidth will exceed the 300 Megabits per second being planned for the Tracking and Data Relay Satellite (TDRS), if raw video data are transmitted for interactive examination by scientists. Another example of high-rate sensors for studies of the sun is the Orbiting Solar Laboratory (at 16 to 20 megabits/sec), being considered for a sun synchronous free-flyer mission. Such data rates cause stresses in all aspects of end-to-end data systems. New technologies and techniques are needed to relieve such stresses. Potential solutions to the massive data problems are: data editing, greater transmission bandwidths, higher density and faster media, and data compression. A combination of all of the above will probably be needed to address the problems completely.

As a step towards studying one particular solution, the Data Systems Technology Working Group, chartered by the NASA Office of Aeronautics and Space Technology, made a recommendation in the summer of 1987 to hold a workshop on scientific data compression. In response to this recommendation a steering committee (see list of members on page 10) was formed with representation from NASA and universities. A meeting of this committee was held on December 14, 1987 to review some of the relevant work in data compression, and define the objectives, agenda and the location of the workshop.

The objectives of the workshop were to:

- Bring together scientists and data compression technologists to better understand science mission requirements and of potential applications of the state-of-the-art data compression techniques to future missions, and

- Formulate guidelines for future data compression research to be supported by NASA for scientific purposes.

With these objectives in view, an agenda was set up to include several invited presentations discussing requirements and constraints, tutorials on compression techniques, descriptions of current research in data compression algorithms and hardware, and case studies of applications. To encourage participation by all the workshop attendees, subpanel discussions were planned, with the attendees required to select from one of four subpanels: Science Payload Operations, Multispectral Imaging, Microwave Remote Sensing, and Science Data Management.

The workshop was held during May 3 through 5, 1988 in Snowbird, Utah. This Proceedings volume summarizes the results of the workshop. Either an abstract and a set of viewgraphs or a short paper is included, for each of the presentations, at the presenter's option. For each of the four subpanels, a separate summary of recommendations has been included later in this volume.

There is a variety of "lossless" and "lossy" data compression methods available today, implemented in both software and hardware. Lossless techniques preserve all the data as collected, including noise and other artifacts, and are fully reversible. Lossy techniques, however, are irreversible in the sense of being able to recover the data as they were originally collected. However, they are still worthy of consideration since they do not necessarily destroy relevant scientific information. In fact, techniques carefully chosen in coordination with scientists may increase the net scientific return from a mission by enabling increased areal or temporal coverage or increased data accuracy. It was acknowledged by the panels that data which need to be preserved in anticipation of unidentified future user requirements have to be stored in losslessly compressed (or uncompressed) form.

There are several applications which can gain by both lossless and lossy compression techniques. This sentiment, and a need for a coordinated effort to identify specific discipline areas and correspondingly appropriate compression techniques, were among the most frequently expressed panel opinions.

David A. Nichols, Jet Propulsion Laboratory
H. K. Ramapriyan, Goddard Space Flight Center
Workshop Chairmen

AGENDA

SCIENCE DATA COMPRESSION WORKSHOP
SNOWBIRD CONFERENCE CENTER
MAY 3-5, 1988

Tuesday, May 3
Superior Room

7:30 - 8:15	REGISTRATION, COFFEE AND DOUGHNUTS	
8:15	OPENING REMARKS - Dave Nichols, JPL/H. Ramapriyan, GSFC	

APPLICATION REQUIREMENTS AND CONSTRAINTS
Session Coordinator: Dave Nichols, JPL

8:30	The EOS Data and Information System	Albert J. Fleig, GSFC
8:55	Space Data Management at the NSSDC	Robert D. Price, GSFC
9:20	The Orbital Maneuvering Vehicle (OMV) Data System	Steve Jones, MSFC
9:45	Video Requirements for Space Station Materials Processing	Charles R. Baugher, MSFC
10:10	BREAK	

COMMUNICATIONS REQUIREMENTS AND CONSTRAINTS
Session Coordinator: Ed Posner, JPL

10:30	TDRS and Advanced TDRS	Aaron Weinberg, GSFC/STI
11:00	Ground Data Networks	John Roeder, NASA HQ/TS
11:30	Deep Space Communication Weather Effects, and Error Control	Ed Posner, JPL
12:00	LUNCH - GOLDEN CLIFF	

COMPRESSION METHODS OVERVIEW
Session Coordinator: Tom Stockham, University of Utah

1:00	Subjective vs. Objective Issues in Data Compression	Tom Stockham, U. of Utah
1:15	Lossless Coding Methods	Robert Rice, JPL
1:45	Lossy Coding Methods	Anil K. Jain, Optivision
2:15	Vector Quantization	Robert Gray, Stanford U.

2:45	BREAK	

DATA COMPRESSION CASE STUDIES
Session Coordinator: Hampapuram Ramapriyan, GSFC

3:00	Noiseless Coding, Some Practical Applications	Robert Rice, JPL
3:20	A Robust Data Compression System for Low Bit Rate Telemetry Test Results with Lunar Data	Khalid Sayood, U. Nebraska
3:40	Data Compression Techniques for Synthetic Aperture Radar	Steve Mango, NRL
4:00	The OMV Video Bandwidth Compression	Garton Lewis, Fairchild Western Systems
4:20	Local Intensity Adaptive Image Coding	Friedrich O. Huck, LaRC
4:40	Data Compression Experiments with TM & Coastal Zone Color Scanner (CZCS) Data	James C. Tilton and H. Ramapriyan, GSFC
5:00	ADJOURN	

5:30 - 7:00 SOCIAL HOUR, NO HOST BAR - MAGPIE ROOM

Wednesday, May 4

8:00 - 8:30 COFFEE AND DOUGHNUTS

SUBPANEL MEETINGS

8:30 - 12:00 CONCURRENT SUBPANEL SESSIONS

ROOM	SUBPANEL TOPIC	SESSION CO-CHAIRMEN
Maybird	Science Payload Operations	Richard C. Hahn, Tom Stockham
Superior A	Multi-Spectral Imaging	Robert D. Price, Richard Weidner
Red Pine	Microwave Remote Sensing	Steve Mango, Steve Wall
Superior B	Science Data Management	Richard Miller, James A. Storer

12:00 - 1:00 LUNCH - GOLDEN CLIFF

DATA COMPRESSION ALGORITHM R&D
Session Coordinator: James A. Storer, Brandeis University

1:00	A Recursive Technique for Adaptive Vector Quantization	Robert Lindsay, Unisys
1:25	Image Compression with Fractals	Michael Barnsley and Alan D. Sloan, Georgia Tech
1:50	Noiseless Compression Using Non-Markov Models	Anselm Blumer, Tufts U.
2:15	Performance of Adaptive Compression on Random Inputs	Martin Cohn, Brandeis U.
2:40	BREAK	

DATA COMPRESSION HARDWARE
Session Coordinator: David J. Eisenman, JPL

3:00	Spaceflight Data Compression Systems for Planetary Mission	David J. Eisenman, JPL
3:10	Alaska SAR Facility Data Compression System	Steve Mango, NRL
3:20	GaAs Implementation of a Limple-Ziv-Welch Compression Algorithm	Mark Hatch, UNISYS
3:40	Design of a Real-Time Vector Quantization Algorithm	Dennis Pulsipher, UNISYS
4:00	A VLSI Chip Set for High Speed VQ of Image Sequences	Richard Baker, UCLA
4:20	Data Compression Standards and Implementation	Barry G. Haskell, AT&T
4:00	Image Processing Using GaAs Technology	Warner H. Miller, GSFC
5:00	ADJOURN	

Thursday, May 5

8:00 - 8:30 COFFEE AND DOUGHNUTS

SUBPANEL MEETINGS

8:00 - 10:40 CONCURRENT SUBPANEL SESSIONS

ROOM	SUBPANEL TOPIC
White Pine	Scientific Payload Operations
Superior A	Multi-Spectral Imaging
Red Pine	Microwave Remote Sensing
Superior B	Science Data Management

10:40 BREAK

SUBPANEL REPORTS
SUPERIOR ROOM

11:00	Scientific Payload Operations	Richard C. Hahn/Tom Stockham
11:20	Multi-Spectral Imaging	Robert Price/Richard Weidner
11:40	Microwave Remote Sensing	Steve Mango/Steve Wall
12:00	Science Data Management	Richard Miller/James Storer
12:20	CLOSING REMARKS	Dave Nichols/H. Ramapriyan

ORIGINAL PAGE IS
OF POOR QUALITY

ACKNOWLEDGMENTS

Many individuals and organizations made significant contributions to the success of the Workshop on Scientific Data Compression. In particular we would like to thank:

the invited speakers, who committed valuable time to give the workshop consistently high quality technical substance;

the sub-panel co-chairmen, some of whom were recruited at the last minute, for organizing and directing the topical discussions and writing summaries of the sub-panel results.;

the session chairmen, who not only put together an excellent technical program, but also managed to keep enthusiastic speakers on schedule;

the participants, who took time from busy schedules, to explore the possibilities of data compression or to understand NASA's diverse needs; many of whom came largely from curiosity, but nevertheless provided the participation and interaction necessary for the workshop success;

the steering group members, who assisted in refining the workshop objectives and agenda;

John Dalton (Goddard Space Flight Center), Wayne Bryant (Langley Research Center) and other members of the NASA Data Systems Technology Working Group, for supporting the idea of a data compression workshop;

Robert Price and Milton Halem (Goddard Space Flight Center) for their encouragement and support;

and the NASA Office of Aeronautics and Space Technology, Information Science and Human Factors Division (Lee Holcomb, Director), which sponsored the workshop and bore most of the workshop expenses.

We would especially like to thank Tom Stockham, University of Utah, for his assistance in helping to assure that the technical participation was of the highest possible caliber and represented the state-of-the-art in data compression technology.

- David A. Nichols
- Hampapuram K. Ramapriyan

SCIENCE DATA COMPRESSION WORKSHOP

STEERING GROUP

John Barker	GSFC
John C. Curlander	JPL
Jeff Dozier	JPL/UCSB
Warner Miller	GSFC
Dave Nichols	GSFC
Eni Njoku	HQ/EE
Ed Posner	JPL
H. Ramapriyan	GSFC
Tom Stockham	Univ./Utah
Jim Storer	Brandeis U.

DATA COMPRESSION WORKSHOP ATTENDEES' LIST

Richard Baker
UCLA 6731 Boelter Hall
Los Angeles, CA 90024
213 825 2468

Michael Barnsley
Georgia Inst. of Tech.
Dept. of Mathematics
Northwest Atlanta, GA 30332

Charles R. Baugher
Marshall Space Flight Ctr ES71
MSFC, AL 35812
FTS 824 7417

Floyd Bednarz
Johnson Space Center Code SE2
Houston, TX 77058
FTS 525 7459

Anderson Bennett
NASA Headquarters Code SSR
Washington, DC 20546
202 453 2786

Joseph L. Bishop
SSPO - ISSPG
10701 Parkridge Blvd.
Reston, VA 22091
703 487 7131

Anselm Blumer
Tufts Univ. Computer Science Dept.
Medford, MA 02155
617 629 2632

Trevor Butlin
Canada Ctr. Rem. Sen.
2464 Sheffield Road
Ottawa, Ontario CANADA, K1A 0Y7
613 952 0762

C.Y. Chang
JPL 4800 Oak Grove Dr.
Pasadena, CA 91109
818/354-9488

Kar-Ming Cheung
JPL 4800 Oak Grove Dr.
Pasadena, CA 91109
818 354 3015

Martin Cohn
Brandeis University
Computer Science Dept.
Waltham, MA 02254

Benjamin V. Cox
Unisys Corporation
640 N. 2200 W.
Salt Lake City, UT 84116-2988
810 594-5287

John C. Curlander
JPL 4800 Oak Grove Dr.
300-235
Pasadena, CA 91109
818 354 8262

David J. Eisenman
JPL 4800 Oak Grove Dr.
156-142
Pasadena, CA 91109
818 354 2744

Waichi Fang
JPL 4800 Oak Grove Dr.
300-329
Pasadena, CA 91109
818 354-1648

Albert J. Fleig
Goddard Space Flt. Ctr.
Code 600.0
Greenbelt, MD 20771
301 286 7747

James Frew
Univ. of California
Computer Systems Lab
Santa Barbara, CA 93106
805 961-8413

Rick Frost
Brigham Young Univ. 459 Clyde Building
Dept Elec&Compu
Provo, UT 84601

Richard Gibby
Unisys
640 N. 2200 West
Salt Lake City, UT 84116
801 594-4771

Tony Gioutsos
Evironmental Res. Inst. of MI
P.O. Box 8618
Ann Arbor, MI 48107
313 994 1200

Rick Golden
Bionetics Corp.
955 L'enfant Plaza
Suite 1500
Washington, DC 20024
202 863 1223

Robert Gray
Stanford University
Electrical Eng. Dept.
133 Durand Bldg
Stanford, CA 94305
415 723 4001

Richard C. Hahn
Rensselaer Polytech. Ins.
Materials Engrg
Troy, NY 12181
518 276 6012

Barry G. Haskell
AT&T Bell Laboratories
Visual Comm Res Dept.
Holmdel, NJ 07733
201 949 5459

Friedrich O. Huck
NASA Langley Research Ctr
MS 473
Hampton, VA 23665
804 865 3777

Anil K. Jain
Optivision Inc.
2655 Portage Bay Ave.
Suite 1
Davis, CA 95616
916 756 4429

Sandee Jeffers
Southwest Research Inst.
6220 Culebra Road
San Antonio, TX 78284
512 522 2010

Anngie R. Johnson
NASA/Reston Office
10701 Parkridge Blvd.
Reston, VA 22091
703 487 7262

Robert V. Jones
Unisys Corp.
640 N. Sperry Way MSF1E03
Salt Lake City, UT 84116
801 594 4353

Steven Jones
Marshall Space Flight Ctr
Mail Code EE63
Huntsville, AL 35812
205 544 4373

Peter B. Kahn
JPL 4800 Oak Grove Dr.
233-208
Pasadena, CA 91109
818 354 8294

Susan Kulas
JPL 4800 Oak Grove Dr.
264-648
Pasadena, CA 91109
818 354 3275

Murat Kunt
Lab./Traitement/Signaux
EPFL-Ecublens (DE)
CH-1015 Lausanne
SWITZERLAND
412 147 2626

Ronald Kwok
JPL 4800 Oak Grove Dr.
156-119
Pasadena, CA 91109
818 354 5614

Jerry Lake
Johnson Space Center
1050 Bay Area Blvd.
SE2
Houston, TX 77058
713 483 7442

Charles Lawson
JPL 4800 Oak Grove Dr.
301-490
Pasadena, CA 91109
818 354-4266

Jun-Ji Lee
JPL 4800 Oak Grove Dr.
111-208
Pasadena, CA 91109
818 354 4993

H. Garton Lewis, Jr.
Fairchild Weston Systems
300 Robins Lane
Syosset, NY 11791
516 349 4949

Robert Lindsay
Unisys Corp.
640 North 2200 West
Salt Lake City, UT 84116
801 594 6616

William D. Lindsay
Unisys Corporation
640 N. Sperry Way
MS F2J05
Salt Lake City, UT 84116
801 594 5312

Kuang Y. Liu
JPL 4800 Oak Grove Drive
156-246
Pasadena, CA 91109
818 354 4835

Irma Lopez
JPL 4800 Oak Grove Dr.
180-704
Pasadena, CA 91109
818 354 6947

Thomas J. Lynch
FBIS
11717 Gregerscroft Rd.
Potomac, MD 20854
703 733 5924

Eric Majani
JPL 4800 Oak Grove Drive
156-246
Pasadena, CA 91109
818 354 9574

Gary K. Maki
University of Idaho
Elect. Engineering Dept.
Mosco, ID 83843
208 885 6045

Anamitra Makur
California Inst. Tech.
Elect. Eng., 230 Steele
116-81
Pasadena, CA 91125
818 356 3729

Stephen Mango
Naval Research Lab.
Code 5381MA
Washington, DC 20375-5000
202 767 2003

Richard Masline
JPL 4800 Oak Grove Dr.
301-490
Pasadena, CA 91109
818 354 4889

Rich Miller
JPL 4800 Oak Grove Dr.
264-648
Pasadena, CA 91109
818 354 8028

Warner H. Miller
Goddard Space Flt. Ctr.
Code 728
Greenbelt, MD 20771
301 286 8183

Rom Narayanswamy
Science & Technology Corp.
101 Research Drive
Hampton, VA 23669
804 865-3777

David A. Nichols
JPL 4800 Oak Grove Dr.
264-648
Pasadena, CA 91109
818 354 8912

Lyndon R. Oleson
US Geological Survey
EROS Data Center
Sioux Falls, SD 57198
605 594 6555

Saumil Patel
Johnson Space Center
1050 Bay Area Blvd.
SE2
Houston, TX 77058
713 483-7442

Ed Posner
JPL 4800 Oak Grove Dr.
264-801
Pasadena, CA 91109
818 354 6224

Robert D. Price
Goddard Space Flt. Ctr.
Code 630
Greenbelt, MD 20771
301 286 9041

Dennis Pulsipher
Unisys 640 N. 2200 W.
M/S F1E03
Salt Lake City, UT 84116-2988
801 594-7074

Hasan Rahman
Johnson Space Center; GE
1050 Bay Area Blvd. SE2
Houston, TX 77058
713 483 7460

Hampapuram Ramapriyan
Goddard Space Flt. Ctr.
Code 636.0
Greenbelt, MD 20771
301 286 8744

Robert Rice
JPL 4800 Oak Grove Dr.
156-152
Pasadena, CA 91109
818 354 2616

William L. Rubin
Unisys Corporation
Unisys Prk, 3333 Pilot Knob
MS U2D17
Eagan, MW 55121
612 456 3655

Khalid Sayood
Univ of Nebraska-Lincoln
Dept./Electrical Engng.
Lincoln, NB 68588
402 472 6688

Mehrdad Shahshahani
JPL 4800 Oak Grove Dr.
238-420
Pasadena, CA 91109
818 354 6334

Alan D. Sloan
Georgia Inst. of Tech.
Dept. of Mathematics
Atlanta, GA 30332
404 894-2700

Philip Sohn
Lewis Research Center
21000 Brookpark Road
M/S 501-6
Cleveland, OH 44135
216 433-3291

Jan J. Sojka
Utah State Univ
Ctr for Atmos Space Science
Logan, UT 84322-4405
801 750 2964

Rodney Spendlove
Unisys 640 N. Sperry Way
Salt Lake City, UT 84116
801 594-5744

Thomas Stockham
Univ. of Utah
Electrical Engng. Dept.
Salt Lake City, UT 84112
801 581 8541

James A. Storer
Brandeis University
Computer Science Dept.
Waltham, MA 02254
617 736 2714

Laif Swanson
JPL 4800 Oak Grove Dr.
238-420
Pasadena, CA 91109
818 354 2757

William Thompson
Analex Corp.
21775 Brookpark Road
Fairview Park, OH 44126
216 779 3784

Loel B. Tibbitts
Unisys Corp.
640 North 2200 West
Salt Lake City, UT 84604
801 594 5181

James C. Tilton
Goddard Space Flt. Ctr.
Code 636.0
Greenbelt, MD 20771
301 286 9510

Steve Wall
JPL 4800 Oak Grove Dr.
183-701
Pasadena, CA 91109
818 354 7424

Richard Weidner
JPL 4800 Oak Grove Dr.
168-522
Pasadena, CA 91109
818/354-2135

Aaron Weinberg
Stanford Telecomms. Inc.
1761 Business Center Dr.
Reston, VA 22090
703 759 1022

Jim Weiss
NASA Headquarters
Code EC
Washington, DC 20546
202 453 1705

Phil Zion
NASA Headquarters
Code EC
Washington, DC 20546
202 453 2140

MICROWAVE REMOTE SENSING SUBPANEL REPORT

Session Coordinators:

Steve Mango
Naval Research Laboratory

Steve Wall
Jet Propulsion Laboratory

Subpanel Members:

Anderson Bennett
C.Y. Chang
Martin Cohn
John C. Curlander
Tony Gioutsos
Robert V. Jones
Ronald Kwok
Charles Lawson
Robert Lindsay
William D. Lindsay
Kuang Y. Liu
Stephen Mango
Philip Sohn
Laif Swanson
Steve Wall

MICROWAVE REMOTE SENSING SUBPANEL REPORT

The Microwave Remote Sensing Subpanel identified three basic categories of needs for data compression by the scientific community in the microwave remote sensing disciplines :

1. to lower the data storage requirements for archiving - especially if the raw data from high volume, high data rate sensors such as SAR's must be archived

2. to ease existing data link limitations and buffer capacities for any data link rate system from the lowest rate network lines to the leading edge, highest link rate systems

3. to fulfill the telescience needs of multiple users with different data requirements.

For cases in the first category where raw data archiving is not required (e.g. an operational SAR), on-board processing, image data compression and/or image classification can be utilized. Indeed, any or all of these procedures should be used wherever a cost-effective procedure which fulfills the specified scientific needs has been or could be developed.

In cases where raw data archiving is desired, lossless and near lossless compression techniques should still be considered at the "front-end" of systems at the Analog-to-Digital (A/D) stage. Compression techniques such as block quantization have been demonstrated to be effective instead of the usual scalar A/D quantization. Vector quantization has been suggested and may indeed produce competitive results but it was felt that the applicability of this technique at this stage of the data stream needed further study.

Data compression techniques that are employed to conserve down-link data in the second category above must consider the intended use of the data. In these cases:

1. the user must have the option to allocate bits to best satisfy his objective over a reasonable range of selectable options, e.g. to allocate bits in a trade-off between swathwidth covered versus quantization level (bits per sample).

2. data compression techniques should be switchable (on or off) so they can be mated to end user requirements.

In the third category many telescience setups could potentially benefit from existing data compression techniques. Whenever data (or imagery) needs to be sent to a remote site, whether it be a field site, a mobile station or a university facility, the opportunities for data compression utilization indicated in the two categories above are also available for the telescience applications. This is the case regardless of the processing stage of the data (or image) that needs to be telemetered.

In the following the results of the Microwave Remote Sensing Subpanel are presented in the form of:

1. specific recommendations

2. statements of the status of various techniques

3. suggested areas of needed research in data compression in the three categories of needs identified.

Recommendation

The philosophy of whether data compression should be employed in a remote sensing system can be approached in several ways.

The Subpanel recommends that the end users' scientific research should probably not directly address whether or not to use data compression but rather the end users' requirements should be used by the mission planners and system designers to evaluate whether or not data compression is needed to fulfill those requirements within the context of the mission technology and budget limitations.

When the mission needs indicate that data compression should be implemented the issue of whether to implement a particular data compression technique should also be driven by:

1. the users' requirements as well as

2. the cost-effectiveness of the technique.

A corollary formulated by the Subpanel was that the principal burden is on the users to define stringently their requirements in terms of the characteristics of the measurables of the microwave sensor.

The concept often used of archiving all the primitive data in order to satisfy in the future as many users as possible and to accomplish this with presently unidentified users' requirements leads to logical traps in developing any specific mission plan as well as in the sensor/platform design for that mission.

Wherever possible, distinct phases of a mission or even separate missions, each driven by identified users' requirements, would be a more wholesome, effective approach to addressing the question of the firm needs for the retention and archiving of the data.

Recommendation

In order to satisfy more diversified users' requirements any implementation of a data compression technique(s) on-board the sensor platform should be switchable (on/off) even if only a low-loss (quasi-reversible) technique(s) is involved.

Recommendation

A lossless data compression technique is essentially a variable length process that depends upon the individual data; therefore, the compression is not as predictable as a fixed length process.

This should be an area of user consideration and an area for further study -- for example, there is an implied trade-off between no data compression versus data compression with increased area coverage or swathwidth. It is strongly recommended that the requirements and impact of Error Correction Coding (ECC) be incorporated in the study.

SYNTHETIC APERTURE RADAR (SAR) DATA COMPRESSION

SAR two dimensional imaging radar systems with their high data rates and data volumes are excellent candidates for data compression applications. This must be emphasized especially for future, planned SAR missions that will incorporate multiple operating frequencies and multiple polarizations with each data channel being acquired at a very high rate.

The coherent nature of SARs gives rise to a rather unique nature for the SAR data. The raw data is not picture-like; it is more like a complex interferogram. The energy of each point target in the scene domain is dispersed over a very large area in the raw data domain. An effective two dimensional matched filtering process can collapse the spread of energy from each point into a two dimensional radar image.

The nature of the SAR data at various stages suggests the possibility of the utilization of data compression at various stages of processing: operating on the raw data, operating on the data after range processing, or operating on the data after complete processing to the image domain.

STATUS: SAR IMAGE DATA COMPRESSION

In the last few years several data compression techniques previously applied to visual imagery have been implemented with two dimensional SAR imagery with promising results.

Recommendation

Conventional data compression methods should be considered as viable techniques to apply to SAR image data. Each method's sensitivities to the special speckle "noise" characteristics of SAR imagery should be better quantified.

STATUS: SAR RAW DATA COMPRESSION

Because the fundamental nature of SAR raw data tends to be "white" and is phase sensitive to small changes in the data, the conventional data compression techniques have not produced comparable results operating on the raw data for the same compression factor as operating on the image data.

Area of Needed Research

Further investigation of the fundamental limitations in the application of data compression techniques to SAR raw data is needed.

This can be a very important issue especially in cases where on-board processing to the image domain is either not desirable or not practical.

Desired Comparisons/Investigations of Data Compression Techniques

Strong opinions in the subpanel and in the SAR community have indicated that both Block Quantization and Vector Quantization techniques might be very useful for the data compression of pre-image SAR data, especially SAR raw data.

A trade-off comparison of the application of Block Quantization and Vector Quantization techniques for pre-image SAR data was considered to be highly desirable.

A second desired investigation was the study of the use of Vector Quantization instead of the commonly used Scalar Quantization at the Analog-to-Digital level for SAR data.

FUNDAMENTAL NATURE OF DATA COMPRESSION

A fundamental tenet of data compression is to produce the most faithful representation of the original data with a reduced data set. When applied to imagery or imagery-like data some data compression techniques intrinsically sacrifice primarily the radiometric fidelity (the intensities), some the geometric fidelity (the spatial representation of the image pixels), while others a hybrid of these two.

Recommendation

The true efficacy of a given data compression technique is often scene dependent and difficult to quantify. However, an attempt should be made to evaluate/categorize each technique offered in terms of its radiometric and geometric consequences in order to serve as a reference framework for users to make a trade-off.

Recommendation

There is often a significant gap in a user's knowledge of a microwave sensor's capabilities and his(her) knowledge of their area of expertise.

Earlier pilot programs to demonstrate the effects of nominated data compression techniques by polling a statistically significant of users should be continued. Each technique should be evaluated for various scene types and for a range of compression ratios. It is strongly recommended that the data compression techniques be evaluated using scientific application criteria; e.g., image classification, rather than the traditional Mean-Square-Error (MSE) criterion.

SUBPANEL OPINION: DATA COMPRESSION TECHNIQUES FOR ON-BOARD AND DATA DISTRIBUTION CENTERS

Technology development and hardware/software implementation of data compression techniques for on-board usage and ground-based central distribution points do not seem to present any fundamental limitations to the usage of data compression.

Further advances in the implementation of data compression techniques in operational systems will require significant user community education with respect to the benefits of data compression, e.g., in the areas of data access, throughput, data volume, and storage requirements.

MULTI-SPECTRAL IMAGING SUBPANEL REPORT

Session Coordinators:

Richard Weidner
Jet Propulsion Laboratory

Robert Price
Goddard Space Flight Center

Subpanel Members:

Richard Baker
Kar-Ming Cheung
Benjamin V. Cox
James Drew
David J. Eisenman
Waichi Fang
Albert J. Fleig
Jun-Ji Lee
H. Garton Lewis, Jr.
Thomas J. Lynch
Eric Majani
Anamitra Makur
Warner H. Miller
Lyndon R. Oleson
Saumil Patel
Ed Posner
Robert D. Price
Dennis Pulsipher
Robert Rice
William L. Rubin
Khalid Sayood
Alan D. Sloan

Jan J. Sojka

James C. Tilton

MULTI-SPECTRAL IMAGING
SUBPANEL REPORT

INTRODUCTION

Within the next two decades NASA will fly or will have flown numerous imaging sensors in the visible and infrared portions of the electromagnetic spectrum. These sensors may be multiple-band spectrometers, such as the HIRIS, or may have only a few bands such as an auroral observer. In turn, the data rate may be massive (> 500 Mbs) or simply large (˜ 1 Mbs). The downlink data channel for most of these missions will not permit transmission of the full sensor data stream. Planetary missions are power limited requiring further transmission restrictions.

Data Compression of some form is required. Data Compression for this report is simply reduction of the number of bits in a data stream. This may be accomplished by a number of means from decimation to universal coding to transform coding to ultimately end-product extraction. Each of the means changes the data stream. However, the methods may or may not be reversible. A reversible compression method implies that a method exists to reconstruct the data stream exactly. Therefore, reversible methods do not lose information or any artifact of the data. However, irreversible compression does not necessarily imply that information is lost. As an alternative distinction, compression methods may be arbitrarily divided by their capability to lose or not lose information. Bear in mind that the term information refers to a probability measure in the sense of communications theory, not necessarily scientific worth. In this sense, some compression schemes may add to the worth.

The most prevalent data compression method is decimation in data quantity via spatial or spectral "editing" or word length reduction. For example, the HIRIS instrument baseline expects to send at most one-half the available spectral bands within a maximum two percent duty cycle. Obviously, this method loses information, but it also represents the easiest and directly cheapest compression method. (Indirect costs may show that this method is not cheap at all.)

There are numerous compression methods relevant to multi-spectral imaging data with varying compression rates and accompanying varying amounts of information loss. This report will examine the "technology" associated with these methods. What compression methods are relevant? What technology represent the state-of-the-art? What new development thrusts are required to utilize a feasible method? What are the implementation issues? What is the scientific utility of the compression technologies? What recommendations can be made for NASA?

RELEVANT TECHNIQUES

Historically, NASA missions have been expensive; returning limited amounts of information about a specific physical area. Great emphasis was placed on preserving all of the information present in the sensor data. Since the observations were usually the first ever, little was known about what portion of the data was important and what was not important. The tradeoff soon resulted in giving up some data in order to receive all of the rest of the data. That is, all of the data would be transmitted that could fit in the data channel.

Using the historical perspective only lossless compression methods are relevant to multi-spectral data. However, the aforementioned scenario is changing. The requirements for the massive capabilities of the new sensors are not necessarily the same as past missions. Analysis of the actual scientific goals and requirements with respect to the capabilities of the lossy compression techniques may show that lossy compression is useful and possibly even mission-enabling. For that reason a variety of lossless and lossy data compression methods are deemed relevant.

The lossless compression methods usually consist of a family of universal source coding algorithms developed at JPL and commonly associated at JPL with Robert Rice. There are new techniques being developed under universal coding. There are also revolutionary techniques such as fractal compression which can be lossy or lossless.

There are many more lossy compression methods. These include the various forms of vector quantization (lattice, finite state, classified, product, predictive, hierarchical, ...), cluster compression, adaptive and fixed block transform coding, DPCM/ADPCM, fractal, information extraction, and of course decimation, to name just a few. The most compression may be achieved with information extraction and/or fractal compression. These methods are very focused toward a specific goal (eg. Hausdorff Distance) and may be viewed as very lossy by other criteria. They may also be computationally massive. However, the vector quantization methods appear to be feasible with moderate computational capabilities while maintaining attractive compression ratios and possibly acceptable information loss. On the other hand, DPCM and the Discrete Cosine Transform are being implemented for use in other domains such as SAR and video compression. Thus, they may be implemented very cheaply on a shared basis. Unfortunately, they also do not get compression beyond approximately 1.8:1 for multi-spectral data.

The acceptance of the lossy and lossless compression methods for future missions rests on several factors including: availability, cost both for implementation and for reversing the compression on the ground (if it needs reversing). Lossy compression methods must also prove that their information loss is acceptable while achieving scientific goals. The discussion of these factors first relies on some assessment of the current state-of-the-art.

STATE-OF-THE-ART

Universal Source Coding (e.g. Rice Lossless Compression) is a mature family of algorithms. It has been flown on Voyager and is planned for Galileo and a host of other planetary missions. BARC and previous Universal Coding algorithms have been implemented for low data rates using 8086 technologies for the planetary missions. Compression factors of 2-3:1 may be reliably achieved with custom MSI electronics at high data rates if the effort is funded.

DPCM is being implemented in GaAs by a number of tasks. The DCT is being implemented with relation to SAR. A custom chip implementation of VQ is in the design stage. The other methods are still largely in the software stage.

Many members of this panel feel that custom VLSI architectures are the only feasible technologies available for compression method implementation. Certainly, the state-of-the-art for flight certifiable electronics is restricted to slow-speed LSI and MSI products in the generic "off-the-shelf" market. Designs using these products yield unsatisfactory results in size, power, performance, and reliability. On the other hand, high speed custom chips such as a GaAs ALU are in the test phase now.

The state-of-the-art is a very fleeting concept in the area of data compression. As stated, decimation is the prevalent method for compression. However, there are proposals before NASA which would advance the threshold significantly if certain implementation issues are satisfied.

IMPLEMENTATION ISSUES

In order to utilize any of these methods, first and foremost the effect of the compression on the data should be quantified using some scientifically meaningful metric. This metric is necessary to gain acceptance by the user community. The metric is very much dependent on the actual implementation of the compression method, as well as on the type of data. In addition, a number of metrics may be necessary for each method in order to relate the effects to the different users.

Secondly, these methods and accompanying implementation must fly on NASA missions. As a result, the technology must minimally use resources of size, weight, and power. The technology must be reliable and thus be space qualified.

The methods must be robust in order to handle a variety of data characteristics. The technology must keep up with the flow of data from the sensor though large data buffers may help somewhat.

Currently, 1.5 to 2 micron CMOS MSI chips are the limit of capability in technology. Few denser chips, custom or otherwise, have been qualified for NASA missions. Furthermore, chip speed may be reduced by as much as 50% for space qualified parts. There is no space qualification standard. Each technology and design must be evaluated for the specifics of each mission. Space qualification for low Earth equatorial orbits is far simpler than for polar orbits through the Van Allen belts where radiation is orders of magnitude greater.

Space qualification is an expensive process. Testing for qualification can cost up to half a million dollars. If the chip does not pass, process modifications may run in the millions. Chip characteristics vary widely from batch to batch. One batch may pass while another may fail.

As mentioned, off-the-shelf electronic parts are currently too slow, too large, and too power expensive for near future missions. An attractive alternative is custom parts using gate arrays or standard cell designs. Unless significant advancements are made in programmable electronic parts, only fixed algorithmic designs may be utilizable in space designs. Thus, every algorithm would have to be implemented in VLSI. Every algorithm change beyond coefficient and data changes would require a new chip. Every new chip would have to be qualified for every mission. At this time these are the only feasible means of meeting mission requirements.

Even with custom parts some algorithms, such as information extraction or fractal compression, require massive computational capabilities and possibly massive memory storage. Giga- operations per second are required to keep up with HIRIS data flow (eighth order bits per second) for moderate calculations such as mixture decomposition.

Gallium Arsenide technology is reputedly naturally radiation hardened. However, only a few chips have been produced so far using GaAs.

Finally, a compression method can not be evaluated until it is flown in actual flight conditions. There is a need to fundamentally adapt many of the algorithms in real time to changing conditions. These changes can only be met by re-configurable or programmable architectures. Fixed architectures do not have the dynamic qualities for such changes. No currently available space qualified architectures can support such changes at the high rates required by the imaging spectrometers.

Rather than accept the limitations raised in the implementation issues at face value, a set of technological thrusts is proposed.

TECHNOLOGY DEVELOPMENT REQUIREMENTS

In order to accept a lossy compression method, a scientist/user must understand the nature of the loss (and gaining in total data). Evaluation of the relevant/irrelevant scientific content rather than communication theory information must be completed and made available to the investigator community. This evaluation can not be performed without extensive interaction with the user community. Often the user community will be too diverse to allow a ready consensus. However, user driven sensors such as HIRIS will allow such interaction for specifically requested information. This interaction is critical to the decimation mode already in the HIRIS baseline. Extending this interaction to allow other forms of compression is necessary as in the Imaging Spectrometer Flight Processor proposal.

A meaningful error metric which may extend beyond mean square error or Hausdorff distance is required to perform the evaluation. Derivation of this user specific metric is a major technological thrust for each of the lossy compression methods.

The lossless compression methods do not have the great requirement for a meaningful metric. However, current implementations of lossless compression do not have the performance required to support the high data rate imaging spectrometers. A space-qualified high-speed custom CMOS or GaAs implementation of lossless compression is required.

Custom VLSI design tools should be pursued. Standard cell libraries and process qualified gate arrays would decrease the chip cost for custom chips. Increased use throughout NASA would build a knowledge base which could be tapped for future missions.

A massive computational though possibly limited-programmable digital signal processor should be pursued as a test bed for data compression algorithms and as a re-configurable compressor. This may be the Configurable High Rate Processor or some high data rate parallel processor such as the information extractor module included in the Imaging Spectrometer Flight Processor. NASA would need to directly support space-related programmable electronics development in addition to utilizing DoD generated parts. Giga-op parallel digital signal processors are required using large 25 ns memories and ultra-high speed support electronics. Such devices can be built using currently available commercial parts. The electronics industry needs support to build space-qualified versions of these devices.

SCIENTIFIC UTILITY

Data compression is advantageous at face value. A more complete data set extends the information available for scientific analysis. However, other circumstances, such as additional ground processing requirements, may temper the benefit. A detailed and individual examination of the benefits versus costs of using data compression is necessary.

There are some general benefits which may be gained from compression. Some forms of compression may perform initial steps of the processing stream. For example, transform coding may be utilized to reduce noise artifacts. Compression may be information extractive. The ultimate compression is the extraction of only the final scientific product. While some of these products require extensive computation, still others, such as band ratios, require only minimal computation capabilities. Compression may extend the observation cycle if combined with prudent processing to perform long term and frequent analysis.

There are various tradeoffs which can be made for the compression utility. The data inherently contains noise. Compression which degrades the signal no more than the already present noise may be lossy in a communication sense but is not lossy with respect to scientific information. A well known technique for image analysis adds white noise to compensate for the observed effect of colored noise. Thus the lossy compression error may be beneficial for some analysis techniques. However, great care should be exercised when allowing additional noise. The observational quality criterion of visual products or speech signals is not usually satisfactory for precise remote sensing analysis. Shifts in position or smoothed spectra may remove precisely that information desired from multi-spectral images.

The science user must be aware that spectral and spatial decimation is a tradeoff which is not always satisfactory either. The requirements of decimation may not allow for frequent observations of limited time phenomena such as auroral storms. The analysis methods may tolerate a noisy data set more than the lack of data during the intensive periods. Decimation is really "lossy" in the broader context.

Overall, the main decision of scientific utility is best made by the scientist users. Comparison of their real goals with the tradeoffs inherent in the data system will decide what compression methods are necessary and useful. Familiarity with available compression methods with an extended set of available compression technologies will help the scientist with that decision.

GUIDELINES FOR OPTIMAL UTILIZATION

The various methods have a host of caveats related to their use. Lossless compression may give you all of the data but you only get roughly 2:1 compression. Lossless compression is ultimately limited by the entropy of the data relative to a model. Models are poor in initial exploration.

Without the technological thrusts mentioned earlier, many of the techniques are not available at all. Even with the necessary technology, some methods such as Fractal compression only work well on truly large data sets, but can maybe get 20:1. Most methods do not work well or at all with random data such as in an encrypted data stream. Vector Quantization works best with moderate to large memory codebooks, but can maybe get 10:1. DPCM and other popular compression methods may not give you compression beyond roughly 1.8:1.

These caveats form a de facto set of rules on using the various methods. A detailed and individual examination of the method's attributes versus the requirements of a data system is necessary.

RECOMMENDATIONS

To summarize, a set of recommendations with regards to data compression technologies and their use may be compiled. First, the data compression methods, lossless and lossy, are considered to be useful to NASA multi-spectral imaging missions. In order to understand their utility, gain acceptance of the technology, and fully utilize the compression potential, the investigator scientists, system design engineers, and compression proponents should work as a team from the very conception of a NASA mission. The sensor and accompanying data system with compression should be jointly designed. Instrument proposals should contain data compression technologies. The instrument and data system should then be reviewed as an overall package. The review team should contain data compression specialists in order to analyze the proposal optimally. The end result is expected to be a better overall use of limited NASA resources such as link bandwidth and power, on-board and ground processing, ground data links, and archiving space.

In order to facilitate evaluation of the compression methods, a possibly new set of error metrics associated with the compression methods should be formulated with respect to individual scientific goals and analysis methods. A set of multi-spectral image models with environmental (atmospheric, lighting, etc.) variables will also facilitate the evaluation. A resource or set of resources, advisory and computational, where a scientist or design engineer can bring his particular data or sensor models to get help in performing that evaluation is strongly recommended. This testbed should also include the capability to evaluate chip sets with the data models. Ultimately, various compression algorithms should be flown in a test environment to illustrate their utility.

NASA missions have varying requirements. For example, deep space exploration spacecraft are severely power, mass, and volume limited. The bandwidth considerations are less important. Small missions have power, mass, and volume limitations similar to deep space explorers. Polar orbiters have more stringent radiation requirements. Data compression technologies which operate well within these constraints should be pursued as vigorously as the more visible high rate sensors.

The available data compression technologies should be extended. Lossless compression should be implemented in custom VLSI, preferably high speed GaAs, but minimally CMOS. Custom VLSI design tools should be pursued throughout NASA to readily and cheaply create the adaptive portion of the lossless technology.

Additionally, lossy compression techniques should be pursued in custom VLSI. A Vector Quantization chip set should be completed in a flight certified architecture.

NASA should sponsor work in high performance programmable or re-configurable VLSI architectures as well as utilizing DoD designs. This work should be pursued both at NASA centers and through established industry. Limited markets are always a problem, but often commercial chips perform the functions required in a NASA shell design. Help in space qualifying a chip set may be an attractive alternative to re-creating the chip from scratch or finding a less useful DoD version.

NASA should pursue synergistic electronics endeavors with DoD. The chip designs should include NASA requirements as well as DoD requirements in order to maximally utilize limited government funding. Often DARPA solely funds and solely guides development efforts with the result that the chip sets may not be useful in similar NASA tasks.

Research and development into lossy compression methods should continue using input from the multi-spectral imaging community. Algorithm development based on error metrics specific to real scientific goals should provide better methods with high compression ratios coupled with minimal resource usage and minimal relevant information loss.

Primers on lossy and lossless compression should be either created or pursued within the NASA community. There are many fine texts already. They should be surveyed on an individual basis and supplemented with sensor/user specific information.

Finally, NASA should continue efforts such as the Data Compression Workshop. The workshop provides a valuable information sharing opportunity in the area of Multi-Spectral Imaging Data Compression.

SCIENCE DATA MANAGEMENT
SUBPANEL REPORT

Session Coordinators:

Richard B. Miller
Jet Propulsion Laboratory

James A. Storer
Computer Science Department
Brandies University

Subpanel Members:

Floyd Bednarz
Joseph L. Bishop
Anselm Blumer
Trevor Butlin
John C. Curlander
Rick Golden
Barry G. Haskell
Sandee Jeffers
Steven Jones
Peter B. Kahn
Gary K. Maki
Richard Masline
Rich Miller
Lindon R. Oleson
Saumil Patel
Hampapuram Ramapriyan
Mehrdad Shahshahani
James A. Storer
William Thompson
Aaron Weinberg
Jim Weiss
Phil Zion

SCIENCE DATA MANAGEMENT
SUBPANEL REPORT

INTRODUCTION

This paper summarizes the meetings of the panel on "Science Data Management" that was held as part of the 1988 NASA Scientific Data Compression Workshop. The panel was co-chaired by the authors of this paper; a list of participants is given in Appendix A. Its purpose was to examine the potential role of data compression in NASA science data management. The discussion started with the question of exactly what should be encompassed by the term "science data management", the subject of the next section. The remainder of this section contains background information on data compression and related terms that are used throughout this paper.

Data compression is the process of encoding ("compressing") a body of data into a smaller body of data. It must be possible for the compressed data to be decoded ("decompressed") back to the original data (or some acceptable approximation to the original data). Not all data is compressible. However, data that arises in practice typically contains redundancy of which compression algorithms may take advantage. Although data compression has many applications, the two most common are the following:

Data Storage: A body of data is compressed before it is stored on some digital storage device (e.g., a computer disk or tape). This process allows more data to be placed on a given device. When data is retrieved from the device, it is decompressed.

Data Communications: Communication links that are commonly used to transmit digital data include cables between a computer and storage devices, phone links and satellite channels. A sender can compress data before transmitting it and the receiver can decompress the data after receiving it, thus effectively increasing the data rate of the communication channel by a factor corresponding to how much the data is compressed.

Effective algorithms for data compression have been known since the early 1950's. There has traditionally been a tradeoff between the benefits of employing data compression versus the computational costs incurred to perform the encoding and subsequent decoding. However, with the advent of cheap microprocessors and custom chips, data compression is rapidly becoming a standard component of communications and data storage. A data compression/decompression chip can be placed at the ends of a communication channel with no computational overhead incurred by the communicating processes. Similarly, secondary storage space can be increased by data compression/decompression hardware (that is invisible to the user).

Some data compression algorithms rely on prior knowledge about the data (statistics about character frequencies, etc.) in order to compress it. By contrast, dynamic or adaptive methods start with no prior knowledge of the data and as more data is seen, more is "learned" about the data, and more compression is achieved.

With lossless data compression, it must be possible to recover an exact copy of the original data from the compressed data. Lossless compression is appropriate for textual data (printed English such as this paper, programming language source or object code, database information, numerical data, electronic mail, certain types of scientific data, etc.) where it may be unacceptable to lose even a single bit of information. By contrast, lossy data compression allows the decompressed data to be an approximation to the original data. For various types of data, what defines a "close approximation" is an area of research in itself. An important application of lossy compression is the compression of digitally sampled analog data (DSAD) such as speech, music, black and white or color images, video, and, of particular significance to this workshop, satellite data. For example, if one sends a digital representation of a photograph over a communication link, it may only be important that the photograph received looks identical to the original (to the human eye); that is as long as this is true, it is acceptable for the bits received to differ from the bits sent. For other applications, what looks good to the human eye may be relatively unimportant compared to criteria based on some other form of measurement.

Although lossless data compression may be viewed as a special case of lossy compression, in practice, textual data and DSAD are two very distinct "flavors" of data, and the amount of compression gained typically differs greatly between the two classes of data. For example, English text may only be compressed by a factor of 2 to 3 by a lossless algorithm where as a black and white photograph that is represented by a 512 by 512 array of 8-bit pixels may be compressed by a factor of more than of 50-to-1 and still "look" acceptable.

With lossy compression, information is intentionally discarded to increase the amount of compression. Information may also be unintentionally lost when errors (often called "noise") occur on the communication link or storage medium. Most "pure" versions of both lossless and lossy compression algorithms assume that all devices in question are noiseless; that is, data received is always identical to that sent. This assumption can be crucial for dynamic methods that rely on an encoder and decoder maintaining certain identical local information (that could become different in the presence of noise). Communication links and storage mediums are never truly noiseless in practice. However, if sufficient error detection and correcting hardware is not available, most data compression algorithms can be modified to tolerate noise.

SCOPE OF THE PANEL

Figure 1 depicts the NASA space data gathering network. Data is first gathered by sensors in space, some preliminary processing may occur at the sensor, some processing (including data compression) may occur at the satellite prior to transmission, data is sent down the satellite link, received data may be processed (including decompression), and the received data is placed in the Level 0 archive. Copies of some of the data stored at Level 0 may be passed through successively higher levels of processing and stored at Levels 1, 2, and 3; see Table 1 for a definition of these levels. Science data sets stored at Levels 0 through 3 consist of measurement data (the actual processed and unprocessed measurements passed down from space), meta data (information about the measurement data), and ancillary data (associated data necessary to use the measurements).

The scope of the panel on science data management was limited to the role of data compression for the storage and distribution of science data sets, after they have reached the Level 0 archive.

The scope of the panel is depicted by the dashed lines in Figure 1. Included in this scope are:

- The handling and storage of science data sets on the ground starting after Level 0 processing.

- Storage of all higher level processed data.

- Buffering of data between the archives and the users.
- The movement of data between points on the ground (including both electronic transfers and shipping by physical media)

ISSUES DISCUSSED BY THE PANEL

After determining the scope of the panel but before attempting to make recommendations for the uses of data compression in science data management, the panel entered general discussion on a number of key issues:

The Necessity of Data Compression

The age-old question is: Are storage costs decreasing so rapidly as to make data compression unnecessary in the near or at least not too distant future? The mere fact that this is an old question lends credit to the hypothesis that no matter how inexpensive storage costs are, there is never enough. The consensus of the panel was that for the next 20 years (and probably much longer), compression is likely to be cost-effective and will not represent a significant complication for storage and transmission hardware and software.

The role of Decompression in Level 0 Processing

The NASA definition of Level 0 processing is the removal of the artifacts of telecommunications. The consensus of the panel was that this processing should include decompression only if it was part of a telecommunication link function that was applied to all data.

The Use of Lossy Compression

The panel agreed that for many classes of data, lossy compression may be necessary to achieve large amounts of compression. However, it was agreed that for permanent archival storage at Level 0 (and in some cases, at Level 1A), only lossless compression methods should be used. Any exceptions to this rule should be carefully studied. The panel agreed that this issue was sufficiently important to warrant a formal statement on the matter. After careful discussion by the panel of the content of this statement, Jim Weiss drafted the following:

It is the goal of Science Data Management to provide losslessly compressed or un-compressed data as the basic set, or lowest level of archival data product maintained by the system. This basic set then would be equivalent to the highest quality of data possible provided by the system and will constitute the permanent archive. In some cases, such as voice or video data, users may choose a form of lossy compression for the archiving of this basic set. In this case the user community must be involved in, and fully aware of, the decision to archive degraded data products with no means for improving that data at a later date.

It is the belief of this panel that the basic archived set need not be lossy compressed, degraded data. This belief is based on the fact that increased densities are being supported by the prime storage media and that the costs associated with these increased densities has been consistently decreasing, thus we are getting more storage for less cost. It is recognized, however, that some users may still choose the lossy case.

Acceptance of Lossy Compression by the Scientific Community

A common reaction of scientists when asked about lossy compression is: "I do not want to loose any information; only lossless compression is acceptable". It was a general consensus of the panel that understanding and acceptance of lossy compression by the scientific community is a key issue. The following items were discussed:

- Much research is needed for measures of quality for decompressed data.

- Scientists need to be educated about the data-quality versus data-volume tradeoff.

- Scientists should be consulted throughout the entire life-cycle of compression algorithms design.

- Algorithms should be designed to provide the user with control over the data-quality versus data-volume tradeoff inherent with lossy compression.

- Data compression hardware and software "tools" must be available to the end-users. Affordability of such tools is an important issue.

APPLICATIONS OF DATA COMPRESSION

Mass Storage: Compression may be used for all data storage. As mentioned earlier, permanent archival storage must, under most circumstances, be lossless. For other kinds of storage, lossy compression may be able to provide large reductions in space while maintaining acceptable quality for the given end-user community. The following points should be noted:

- The use of lossy compression must be studied on an application specific basis.

- In all cases, the data compression/decompression algorithms must be stored with the data.

- Compression algorithms should be designed so as to insure that a single error cannot propagate additional errors over large amounts of data.

- Algorithms and protocols should be ISO compatible where feasible.

Electronic Movement of Data: The two major types of electronic movement of data are browsing and product delivery. For the browse application, highly lossy techniques that yield high data-rates may be the most appropriate. For product delivery, larger time delays may be acceptable to insure adequate quality. Algorithms that allow the user to select a given quality versus data-rate tradeoff (which in the limit could force lossless compression) are most appropriate for this application. The following points should be noted:

- Decompression algorithms that are consistent with low-cost work-stations are needed.

- Compression/decompression algorithms must be designed to work with noisy links. The effective data-rate should degrade "gracefully" as the noise on the link increases.

On-Line Storage: Most of the issues discussed above for mass storage and electronic movement apply here. An additional issue that was discussed by the panel is the use of data compression in conjunction with data base management systems; the consensus was that this issue requires future research.

RECOMMENDATIONS FOR FURTHER STUDY AND DEVELOPMENT

The general conclusion of the panel was that data compression can most surely play a significant role in science data management, even if only existing technology is employed. However, to fully realize the potential of data compression, the panel recommends the following research be encouraged and funded by NASA and other agencies:

- Further development of high performance data compression algorithms, software, and hardware.

- Measures for data quality for end-users.

- Development of valid data descriptions (instrument and application dependent) that could be used to enhance compression algorithms for particular applications.

- Compression algorithms applicable at the data representation level.

- Study of whether compression should be applied at the format level.

- Compression techniques for information stored and searched using data base management systems.

- Efficient methods of combining compression with encryption and error correction/detection.

- A periodic survey of the state of the art of data compression.

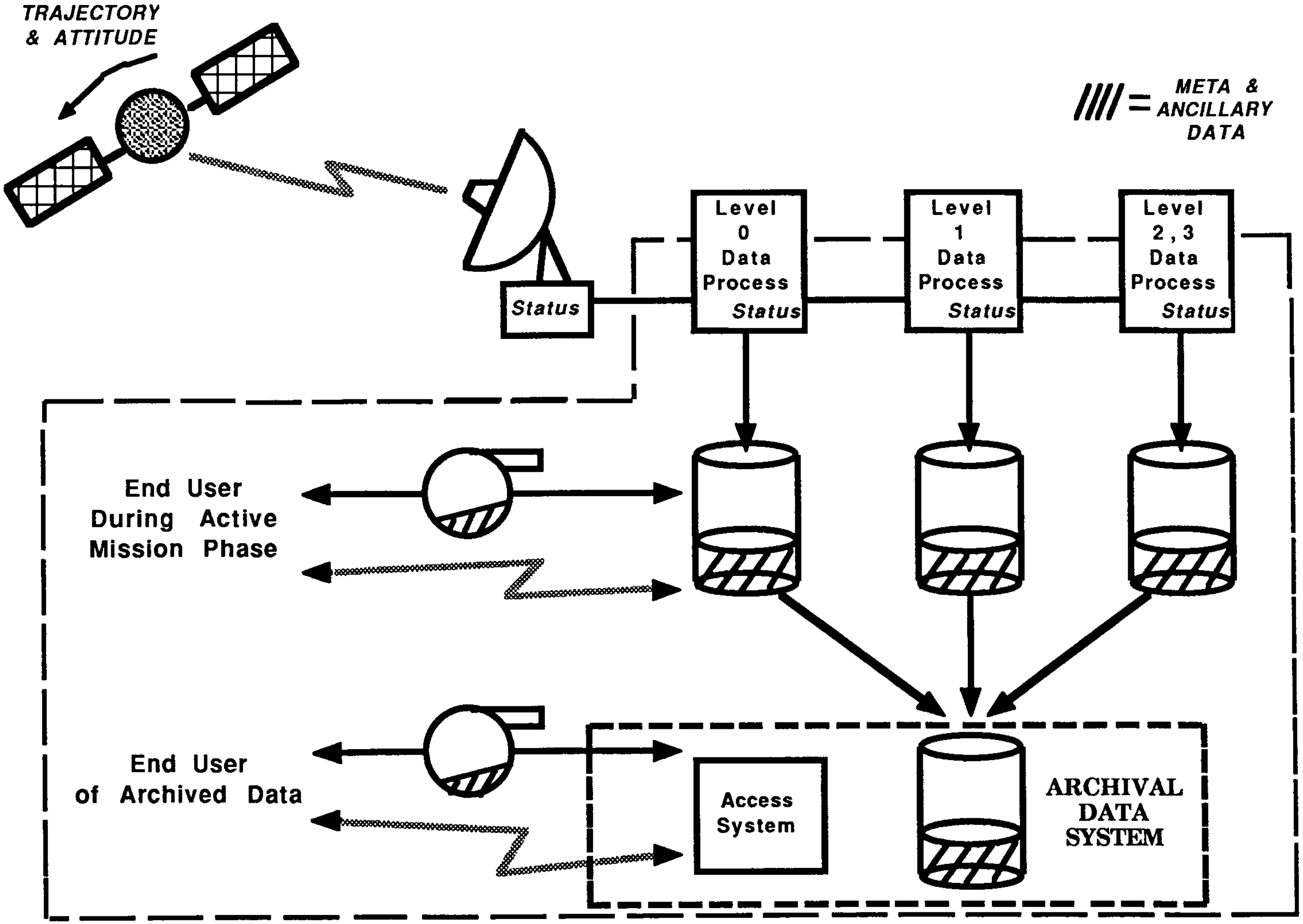
Finally, in addition to the above items, the panel recommends that NASA continue to sponsor a periodic workshop or conference on data compression.

TABLE 1. DATA LEVEL DEFINITIONS

The following "level" definitions are used to describe the various stages of data processing which occur on remote scientific data.

Level 0	Raw instrument data at full resolution where the artifacts of the transport of the data from space to ground have been removed but no further processing has taken place.
Level 1A	Reconstructed, unprocessed instrument data at full resolution, time-referenced, and annotated with ancillary information including radiometric and geometric calibration. The exact definition of 1A varies with science discipline; however, the processing that produces 1A data is always reversible to recreate the original raw measurements.
Level 1B	Data that has been processed to sensor units (i.e., radar backscatter cross-section, brightness temperature, etc.). Not all instruments have a Level 1B equivalent. May be an irreversible product.
Level 2	Derived physical variables (e.g. ocean wave heights, soil moisture, surface temperature) at the same resolution and location as the Level 1 data.
Level 3	Variables mapped on uniform space-time grid scales. Sometimes with completeness and consistency properties (e.g. missing points interpolated, mosaicing to form larger regions).
Level 4	Model output or results from analyses of lower level data (i.e., variables that were not measured directly by the instruments, but instead are derived usually from a combination of instruments' measurements).

Science Data Management: Definition of Scope


TRAJECTORY
& ATTITUDE
//// = META & ANCILLARY DATA
Status
Level 0 Data Process Status
Level 1 Data Process Status
Level 2,3 Data Process Status
End User During Active Mission Phase
End User of Archived Data
Access System
ARCHIVAL DATA SYSTEM

SCIENCE PAYLOAD OPERATIONS
SUBPANEL REPORT

Session Coordinators:

Thomas G. Stockham, Jr.
University of Utah

Richard Hahn
Rensselaer Polytech Institute

Subpanel Members:

Michael Barnsley
Charles R. Baugher
Joseph L. Bishop
Rick Frost
Richard Gibby
Robert Gray
Richard C. Hahn
Friedrich O. Huck
Anil K. Jain
Anngie R. Johnson
Peter B. Kahn
Murat Kunt
Jerry Lake
Rom Narayanswamy
David A. Nichols
Hasan Rahman
Rodney Spendlove
Thomas Stockham
Loel B. Tibbitts

SCIENCE PAYLOAD OPERATIONS SUBPANEL REPORT

The Science Payload Operations Subpanel limited its attention almost exclusively to Earth-orbiting vehicles, such as the space shuttle and the space station. The focus of this panel was not the acquisition of data in space-based remote sensing, but rather the operations of equipment and experiments in space for scientific purposes. The high rate data needs for on-board laboratory-type experiments are typically confined to video for visual inspection and do not constitute precise measurements. This means that lossy forms of data compression are typically acceptable. For study purposes, the spectrum of science payload operations was divided into three parts by the panel. They are:

a. Material Science
b. Laboratory Support
c. Telerobotics/Telescience/Teleoperations

The major subcategories in material science were: (1) combustion, involving 35 experiments, (2) fluid dynamics, involving approximately 12 experiments, and (3) containerless processing and furnace facilities. Laboratory support includes glove box and workbench operations, and on-board image buffering. Telerobotics encompasses activities such as orbital-maneuvering vehicles, planetary rovers, crew training via uplinks, and teleconferencing.

The panel determined that there are compression needs for three facets of space operations:

a. Command and control
b. Data transfer
c. Onboard data storage

These three categories respectively address: (1) the problem of examining experiments from the ground as they are being performed, (2) transfer of raw data to the ground for processing, and (3) in-space storage of massive data for transmission to the ground at a later time. Through inputs from various sources and discussions during the sessions, it became clear that the type and degree of compression required depends very much on the type of experiment in question.

A key question arose as to whether compression ability should be provided by the space station as a common service or by the experiment itself. It was decided that both approaches were needed to be able to embrace the broad spectrum of requirements effectively. It seems certain that the space station will need to provide a basic compression facility operating perhaps at a ratio of four to one. Individual experiments would then have the option of employing lossless or lossy compression, as additionally required.

Because of the broad variety of experiments which are expected in space, both real-time and delayed transfer of data from the space-based platform to the ground will almost certainly be required. In many cases this will involve very large storage capacities. Indeed, some preliminary experimental objectives lead to storage requirements which appear unachievable at this point. Nonetheless, on the average, data rate problems seem to be more serious than data volume problems.

Because of the critical nature of these issues, the subpanel suggests that a formal sponsorship by NASA of further work in data compression techniques, primarily for the support of space station experiment designers, should be undertaken, starting immediately. As a part of this activity, it is recommended that NASA obtain, through its own resource and with the help of session participants, a broad list of compression researchers capable of participating with or assisting NASA in these tasks.

The panel recognizes that space science experimenters, because of their disciplines, may need some assistance in correlating the technologies of compression with practical and theoretical requirements of their proposed experiments. As a result, the subpanel took upon itself to begin the process of generating a matrix of compression types and capabilities. The matrix (attached) addresses five types of levels of compression from lossless to feature extraction. These are cross-referenced with eight capabilities embodying both practical and theoretical parameters. The chart is completed in the areas where the panel reached general consensus. The areas left blank are either unknown or too complex to have a meaningful generalization. It is believed that a fully-completed and documented guide of this sort would aid participants in making appropriate choices concerning NASA missions.

The panel spent some time considering high impact technology areas which would greatly ameliorate data transmission and storage problems for NASA in the future. Two of these seemed urgent enough to mention here. The first technology area involves integrating coder and sensor hardware, perhaps right at the focal plane, in high-performance applications where the hardware proliferation implied by nonintegration would result in too great a performance penalty. The second technology area has to do with mass storage. It is clear that the data rate imposed by some advanced experiments, especially in the combustion area, utilizing high resolution/high frame rate video is far too great to permit either direct transmission or data storage. The reason for the latter is the sheer bulk of data implied. While very large data storage capacities are now emerging, especially in the areas of optical disk and optical tape, it would seem prudent for NASA to examine the relevance of data compression in enhancing the performance of these devices in order to meet its future unique mission requirements.

		LOSSY	LOSSY	LOSSY	FEATURE
	LOSSLESS	PREDICTIVE	TRANSFORM	VQ	EXTRACTION
COMPRESSION FACTOR RANGE	2:1-4:1	2:1-6:1	4:1-16:1	4:1-50:1	10:-100:1
LEAD TIME (years for flight qualification)	2	2	4	6	5
POWER, WEIGHT, COST					
DISTORTION TYPE, DEGREE	NONE				
SENSITIVITY TO CHANNEL ERRORS					
CHANNEL DECAY	PIXELS	PIXELS	10s OF PIXELS	LINES	FRAMES
FLEXIBILITY	HIGH				
MAX INPUT DATA RATE (Current, future)					

APPLICATION REQUIREMENTS

AND

CONSTRAINTS

DAVE NICHOLS
JET PROPULSION LABORATORY
SESSION COORDINATOR

THE EOS DATA AND INFORMATION SYSTEM

Albert J. Fleig
Goddard Space Flight Center

WHAT DATA COMPRESSION WILL DO

- ON BOARD
    - HIGH SPEED NETWORK
    - TAPE RECORDERS
        - SIZE
        - RATE
    - MULTIPLEXER, Xfr FRAME

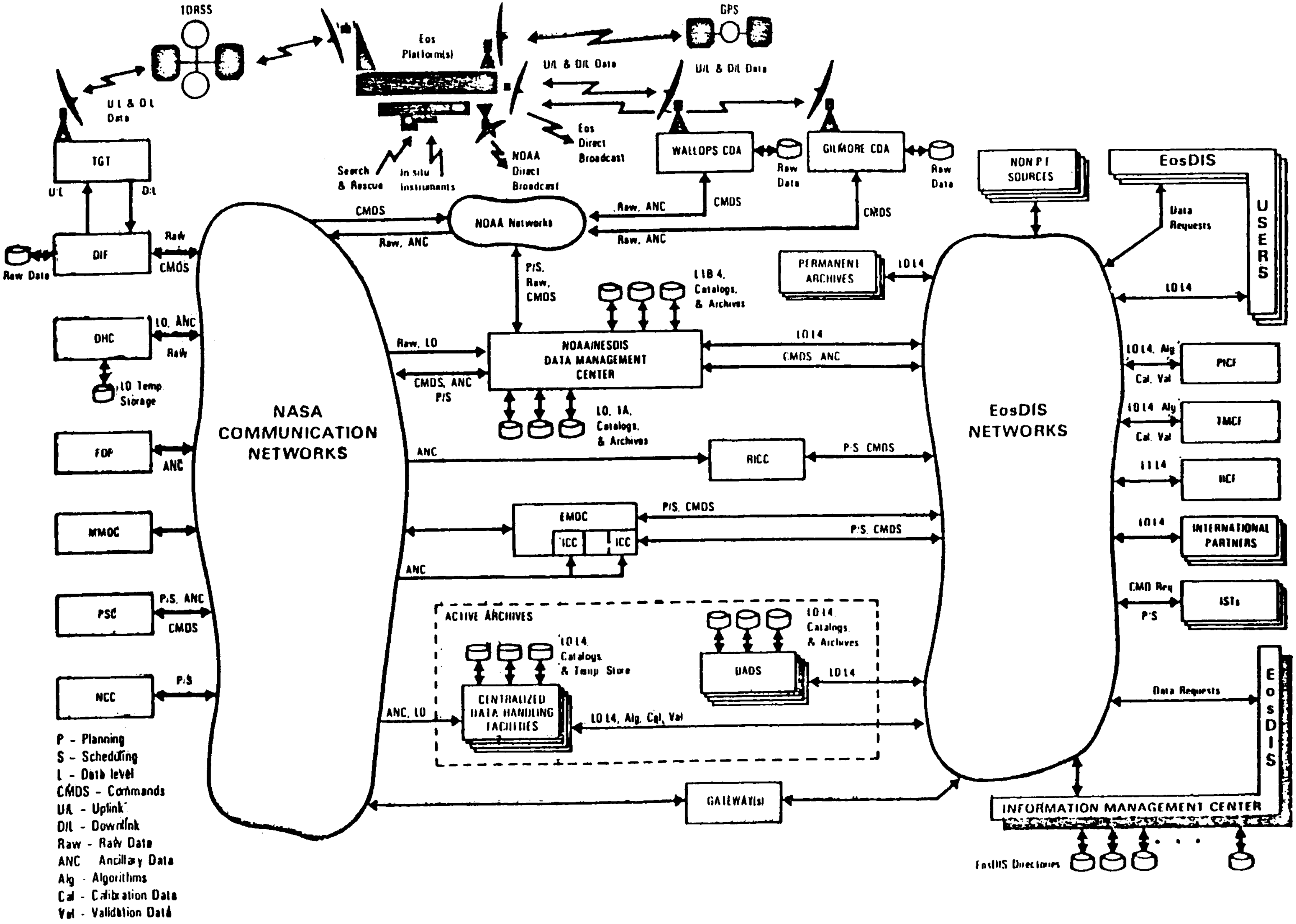
- DOWNLINK
    - SIMPLIFY SCHEDULING
    - REDUCE TIME

- DIF/DHC
    - SMALLER COMPUTERS
    - LESS STORAGE
    - EASIER TRANSMISSION

- L1-4 PROCESSING
    - LESS COMPUTE
    - LESS STORAGE
    - REDUCE OPERATING COSTS
    - EASIER TRANSMISSION

- ARCHIVAL
    - LESS COMPUTE (SUBSETS, PRODUCTS)
    - LESS STORAGE
    - REDUCE OPERATING COSTS
    - EASE DELIVERY
    - EASE MAINTENANCE

- SCIENCE USER
    - REDUCE COST (?) NOT FOR HIGHER LEVEL
    - EASE INGEST (?) NOT FOR HIGHER LEVEL
    - REDUCE PROCESSING LOAD NOT FOR HIGHER LEVEL



PRELIMINARY EOS DIS ARCHITECTURE

ORIGINAL PAGE IS
OF POOR QUALITY

OVERVIEW: APPROXIMATE INSTRUMENT DATA REQUIREMENTS

NPOP - 1 P.M.

INSTRUMENT	SUPPLIER/ CLASS	OPERATING DATA/RATE MBPS (AVG/PK)	ESTIMATED DUTY CYCLE		ESTIMATED DAILY VOL. TERABITS	LINK		DATA PROCESSING LEVEL RESPONSIBILITY				
			%D	%N		TDRSS	DB	0	1	2	3	4
ALT-1	N; OPS	.008/.013	100	100	.0007	T	DB	N/T	N	i	i	i
AMRIR	N; OPS	5.5	100	100	0.475	T	DB	N/T	N	i	i	i
AMSR *1	J; RES	.01	100	100	0.0009	T	--	T	J	i	i	i
AMSU (2)	N; OPS	.0065	100	100	0.0005	T	DB	N/T	N	i	i	i
ARGOS +	F: OPS	.0025	100	100	.0002	?	DB	N	F	i	i	i
CR	AO; RES	.001	100	100	<.0001	T	--	T	DC	i	i	i
ERBI (NS)	N; OPS	.001	100	100	<.0001	T	DB	N/T	N	i	i	i
ERBI (S)	FRG;OPS	.001	100	100	<.0001	T	DB	N/T	N	i	i	i
GOMR *2	N,C;OPS	.01	100	100	0.0009	T	DB	N/T	N	i	i	i
HIRIS *3	U; FAC	10/280	2	0	0.86	T	--	T?	DC	DC/i	DC/i	i
ITIR *4	J; FAC	8.3/52	20	20	0.72	T		T	DC	DC/i	DC/i	i
MAG	AO; RES	.002	100	100	0.0002	T	--	T	DC	i	i	i
MERIS *5	E; FAC	2.4/4.5	100		0.21	T		T	DC	DC/i	DC/i	i
MODIS-N *6	U; FAC	4.85/8.3	100	0	0.42	T	--	T	DC	DC/i	DC/i	i
MPD	AO; RES	.001	100	100	<.0001	T	--	T	DC	i	i	i
PPS/PODS	AO; RES	<.001	100	100	<.0001	T	DB	N/T				
S & R	I; OPS	.0001	100	100	0.00001	-	DB	I				
SCATT-1	N; OPS	.0032	100	100	0.0003	T	DB	N/T	N	i	i	i
SEM	N; OPS	.003	100	100	<.0003	T	DB	N/T	N	i	i	i

* SEE NOTE PAGE

T = CDOS I = INTERNATIONAL DC = DATA CENTER i = INVESTIGATOR

OVERVIEW: APPROXIMATE INSTRUMENT DATA REQUIREMENTS

EPOP - 1 A.M.

INSTRUMENT	SUPPLIER/ CLASS	OPERATING DATA/RATE MBPS (AVG/PK)	ESTIMATED DUTY CYCLE		ESTIMATED DAILY VOL. TERABITS	LINK		DATA PROCESSING LEVEL RESPONSIBILITY				
			%D	%N		DR	DB	0	1	2	3	4
ALT-2	E; P/O	.013	100	100	0.001	DR	?	E	E			
AMIR	IT.;RES	.1	100	100	0.009	DR	--	E	IT	i	i	i
AMRIR	N; OPS	5.5/5.5	100	100	0.475	--	DB	E/N	E			
AMSU (2)	UK; OPS	.0065	100	100	0.0006	-	DB	E/N	UK			
ARGOS +	F: OPS	.0025	100	100	0.0002	--	DB	E/N	F			
ATLID	E; RES	.01	50	50	0.0004	DR	--	E	E	D	i	
ATSR +	AUS;OPS	.213	100	100	0.184	--	DB	E/N	AUS			
CSR	FRG;P/O	.008	100	100	0.0007	DR	?	E	FRG			
GLRS	U; RES	.09/.5	15	15	0.008	DR	--	E/T	E/DC	DC/i	DC/i	
HRIS *10	E; RES	39/200	20		2.2	DR	--	E	E			
MAG	AO; RES	.002	100	100	0.0002	DR	--	E	E/DC			
MPD	AO; RES	.001	100	100	<.0001	DR	--	E	E/DC			
PPS/PODS	AO; P/O	.0001	100	100	0.00001	DR	?	E	E			
S & R	I; OPS	.0001	100	100	0.00001	--	DB	I				
SAR-C *11	E; RES	75/200	50	25	6.5	DR	--	E	E	E	E	E
SCATT-2	J; P/O	.003/.005	53	53	0.0003	DR	?	E/N	E			
SEM	N; OPS	.003	100	100	0.0003	--	DB	E/N	E			

* SEE NOTE PAGE

T = CDOS I = INTERNATIONAL DC = DATA CENTER i = INVESTIGATOR

REV-3 8/10/87

OVERVIEW: APPROXIMATE INSTRUMENT DATA REQUIREMENTS

NPOP - 2 P.M.

		OPERATING	ESTIMATED		ESTIMATE			DATA PROCESSING LEVEL				
	SUPPLIER/	DATA/RATE	DUTY	CYCLE	DAILY VOL.	LINK			RESPONSIBILITY			
INSTRUMENT	CLASS	MBPS (AVG/PK)	%D	%N	TERABITS	TDRSS/	DB	0	1	2	3	4
F/P-INT	AO; RES	.005	100	100	0.0004	T	--	T	DC	i	i	i
IR-RAD *7	AO; RES	.004	100	100	0.0003	T	--	T	DC	i	i	i
MAG	AO; RES	.002	100	100	0.0002	T	--	T	DC	i	i	i
MLS	AO; RES	.30	100	100	0.0026	T	--	T	DC	i	i	i
MPD	AO; RES	.001	100	100	<.0001	T	--	T	DC	i	i	i
PEM	AO; RES	.004	100	100	0.0003	T	--	T	DC	i	i	i
SAR *8	U,I;RES	20.3/280	50	50	1.754	T/	DB	T?/C	DC/i	DC/i	DC/i	
SUB-MM *9	AO; RES	.003	100	100	<.0003	T	--	T	DC	i	i	i
SUSIM	AO; RES	.002	100	0	<.0002	T	--	T	DC	i	i	i

* SEE NOTE PAGE

T = CDOS I = INTERNATIONAL DC = DATA CENTER i = INVESTIGATOR

OVERVIEW: APPROXIMATE INSTRUMENT DATA REQUIREMENTS

JPOP - 1 A.M.

INSTRUMENT	SUPPLIER/ CLASS	OPERATING DATA/RATE MBPS (AVG/PK)	ESTIMATED DUTY CYCLE		ESTIMATED DAILY VOL. TERABITS	LINK DR/DB		DATA PROCESSING LEVEL RESPONSIBILITY				
			%D	%N				0	1	2	3	4
OCTS	J; RES	/2.1				DR	?	J				
AVNIR	J; RES	/60				DR	?	J				
LAWS	U; RES	.05/.1	50	50	.0043	DR	?	J				
AMSR	J; RES	0.1	100	100	.001	DR	?	J				
(SAR-L)	J; RES					DR	?	J				
SAR-X	J; RES	/200				DR	?	J				

INSTRUMENT DATA REQUIREMENTS OVERVIEW

REV 3 8/10/87

* NOTES

*1	AMSR	ASSUMES INTELSAT LINK (TBD)
*2	GOMR	TWO INSTRUMENTS, TOMS - TOTAL OZONE MONITOR GLS - GLOBAL LIMB SCANNER
*3	HIRIS	AVERAGE RATE IS 10 MBPS, LONG TERM AVERAGE IS APPROXIMATELY 3 MBPS
*4	ITIR	ASSUMED AVERAGE RATE
*5	MERIS	SOME OCEAN DATA MAY BE VIA DB: TBD
*6	MODIS-N	DAY RATE 8.4 MBPS, NIGHT RATE 1.5 MBPS
*7	IR-RAD	ESA INSTRUMENT IS THE CSLR
*8	SAR	MAPPING MODE @ 100 MBPS; RESEARCH MODE @ 300 MBPS
*9	SUB-MM	ESA INSTRUMENT IS THE FIRE
*10	HRIS	PLATFORM DATA VOLUME MAY BE LIMITED TO $1x10^{12}$ BITS/DAY TOTAL
*11	SAR-C	PLATFORM DATA VOLUME MAY BE LIMITED TO $1x10^{12}$ BITS/DAY TOTAL

ADDITIONAL NOTES

SAR ON ESA PLATFORM IS A SINGLE FREQUENCY SAR; DATA RATE QUOTED MAY BE HIGHER.

INSTRUMENT ASSIGNMENTS REFLECT THE INTERNATIONAL CO-ORDINATION MEETING (OTTAWA, MAY 87) AND NOT THE AUGUST 1986 BASELINE.

DATA RATES AND VOLUME ARE ESTIMATES ONLY; A DATA RELAY SATELLITE IS ASSUMED FOR THE ESA AND THE JAPANESE PLATFORMS.

ALTHOUGH CSLR AND FIRE ARE BOTH LISTED, IT MAY BE UNLIKELY THAT BOTH WILL BE PROVIDED.

JPOP-1 INSTRUMENT DATA NOT YET AVAILABLE.

LEGEND

AO	SUPPLIED UNDER AO; P.I. CLASS
AUS	AUSTRALIA
C	CANADA
DB	DIRECT BROADCAST
DR	DATA RELAY SATELLITE (ESA)
E	ESA
F	FRANCE
FAC	FACILITY INSTRUMENT (SAME AS CORE INSTRUMENT)
FRG	FEDERAL REPUBLIC OF GERMANY
I	INTERNATIONAL
J	JAPAN
N	NOAA
OPS	OPERATIONAL
P/O	POTENTIAL OPERATIONAL INSTRUMENT
RES	RESEARCH INSTRUMENT
U	U.S.A.
U.K.	UNITED KINGDOM

LIST OF INSTRUMENT ACRONYMS USED IN ANNEX 1

ALT	Radar Altimeter (TOPEX Class of instrument)
AMSR	Advanced Microwave Scanning Radiometer (1500 Km swath; 5 bands (5-60 GHz); 2-20 Km resolution)
AMSU	Advanced Microwave Sounding Unit (as on NOAA - K,L,M)
ARGOS+	An advanced version of the Data Collection and Localization System of NOAA - K,L,M
ATLID	Atmospheric Lidar (a backscatter lidar)
ATSR+	An advanced version of the Along Track Scanning Radiometer provided for ERS-1
AVHRR	Advanced Very High Resolution Radiometer (as on NOAA-K,L,M)
CLSR	Cooled infra-red Limb Sounding Radiometer
CR	Correlation Radiometer (a gas cell non-dispersive spectrometer)
CSR	Conical Scan Radiometer (for measuring Earth radiation budget)
Direct	A direct downlink similar to HRPT, APT, and DSB on NOAA - Broadcast K,L,M
FIRE	Far Infra-Red Experiment (a submillimetre wave limb sounder)
F/P-INT	Fabry-Perot Interfermeter
GLRS	Geodynamic Laser Ranging System
GOMR	Global Ozone Monitoring System
HIRIS	High Resolution Imaging Spectrometer (as propose by NASA)
HIRS	High Resolution Infra-red Radiation Sounder (as on NOAA - K,L,M)
HRIS	High Resolution Imaging Spectrometer (as proposed by Europe)
MAG	Magnetic Field Monitor

MERIS	Medium Resolution Imaging Spectrometer (for ocean and coastal ocean monitoring)
MLS	Microwave Limb Sounder
MODIS-N	Moderate Resolution Imaging Spectrometer - Nadir looking
MODIS-T	Moderate Resolution Imaging Spectrometer - Tiltable
MPD	Magneto-Plasma Dynamics
OCTS	Ocean Color and Temperature Scanner
PEM	Particle Environment Monitor
PPS/PODS	Precise Positioning System/Precise Orbit Determination System
SAR	Synthetic Aperture Radar
SEM	Space Environment Monitor
S & R	Search and Rescue (as on NOAA - K,L,M)
STP	Solar Terrestrial Physics
SUSIM	Solar Ultraviolet Spectral Irradiance Monitor
WIND-SCATTEROMETER	Microwave Wind Scatterometer; double sided swath

SPACE DATA MANAGEMENT AT THE NSSDC: APPLICATIONS FOR DATA COMPRESSION

James L. Green
National Space Science Data Center

ABSTRACT

The National Space Science Data Center (NSSDC), which was established in 1966, is the largest archive for processed data from NASA's space and Earth science missions. The NSSDC manages over 120,000 data tapes with over 4,000 data sets. The size of the digital archive is approximately 6,000 gigabytes with all of this data in its original uncompressed form. By 1995 the NSSDC digital archive is expected to more than quadruple in size reaching over 28,000 gigabytes.

The NSSDC is beginning several new thrusts allowing it to better serve the scientific community and keep up with managing the ever increasing volumes of data. These thrusts involve managing larger and larger amounts of information and data online, employing mass storage techniques, and the use of low rate communications networks to move requested data to remote sites in the United States, Europe and Canada. The success of these new thrusts, combined with the tremendous volume of data expected to be archived at the NSSDC, clearly indicates that new and innovative storage and data management solutions must be sought and implemented.

Although not presently used at the NSSDC, data compression techniques may be a very important tool for managing a large fraction or all of the NSSDC archive in the future. Some future applications would consist of compressing online data in order to have more data readily available, compress requested data that must be moved over low rate ground networks, and compress all the digital data in the NSSDC archive for a cost effective backup that would be used only in the

event of a disaster.

INTRODUCTION

The purpose of the NSSDC is to serve as an archive and distribution center for data obtained on NASA space and Earth science flight investigations and to perform a variety of services to enhance the overall scientific return from NASA's initial investment in these missions. The NSSDC usually receives data from NASA principal investigators or directly from NASA projects where facility instruments are being flown. However, the NSSDC also obtains data from other government and international agencies involved in space research.

Although the NSSDC does not currently store the data it manages in its archive in compressed form, data compression may very well be a future requirement. In this paper I will discuss the reasons for considering the use of data compression techniques at the NSSDC by looking at the future requirements for data distribution and the growing size of the data center's archive. I will concentrate on the situation at the NSSDC but it must be recognized that many other institutions, universities and other government agencies are in a similar situation.

CURRENT AND FUTURE NSSDC ARCHIVE

The NSSDC archives and manages both digital and non-digital data. The digital archive is stored on approximately 120,000 magnetic tapes with the volume of over 6,000 gigabytes. There are over 4,000 data sets that are supported with appropriately 250 new data sets being archived per year. The most requested digital data sets are stored at the NSSDC (comprising about 35,000 tapes) with the remainder of the archive stored in the Federal Records Center (FRC) about 20 miles away. In addition to the digital archive, the NSSDC has a photo or film archive of over 2 million feet of film, 41,000 sheets of microfiche, and 39,000 microfilm roles. The charge for obtaining data from the NSSDC is usually for replacement costs in materials and supplies (example: a blank tape or equivalent is needed for one tapes worth of archived data).

From the time it was established in 1967 until 1985 all requests for NSSDC held data were in the form of letters or phone messages which was consistent with the "offline" management of the data that was employed. Requests for offline archived data typically takes 2 weeks if the data is held locally at the NSSDC. If the needed data is in the FRC, the request will take a month or more to be satisfied. This situation is labor intensive and involves interacting with another federal organization. Currently, the NSSDC must accumulate requests for data stored in the FRC and makes two trips per month to obtain the data.

In 1985 NASA was acquiring approximately 360 gigabytes of data per year. Assuming both currently approved and most likely approved NASA missions, the acquired data volume by 1995 will reach well over 2,400 gigabytes per day[1]. This is a staggering rate. Figure 1 shows the data volumes per scientific discipline that have been archived and are expected to be archived at the NSSDC. As discussed above, the NSSDC has currently about 6,000 gigabytes of data. The size of the archive past 1988 is a projection and considers the arrangements being made with the NSSDC and the missions that are currently approved. If this projection holds true, then by 1995, the NSSDC will have over 28,000 gigabytes in its archive.

The physical space that the NSSDC has to manage is nearly full, both locally and at the FRC. From the predicted amount of data to be archived, as shown in Figure 1, the NSSDC must implement mechanisms to store data on higher density media (by a factor of 5 to 10) and/or implement data compression techniques. At the NSSDC, use of data compression techniques as a routine mechanism to pack data on media can only be a viable mechanism when it becomes accepted and is in wide spread use in the scientific community. This acceptance is occurring (see section, Networking of AVHRR Data), but only very gradually.

ARCHIVE SAFE STORAGE

A national archive needs to have operational plans for insuring that a natural disaster, such as fire, does not permanently destroy irreplaceable data. For a data center, where most of the archive is in digital form, then a copy of the data stored in another location would be the best solution for safe storage of archived digital data.

In the case of the NSSDC, over the last 20 years, it has received data for archiving from investigators at hundreds institutions across the country. In this situation, the remote investigators retain the original data with a copy sent the NSSDC. Due to budget constraints and inadequate resources to provide a complete backup, the NSSDC's disaster recover plan is to request a copy of the data from the original producers. These plans are inadequate as a viable disaster recovery plan since many of the investigators would not be able to reproduce the data that is more than four or five years old for a variety of reasons (ex., inadequate resources, older tapes written at low density formats, etc). In addition, NASA's missions are now moving toward facility instruments where the NSSDC must assume full responsibility for the data being archived since it is coming from a short lived project with resources that are usually just adequate to keep up with the new incoming data with little or no reprocessing possible.

Plans are now being devised at the NSSDC, that once in place, will provide for a complete safe storage as a backup of the NSSDC digital archive. With such large volumes of data to back up, a cost effective solution requires an extremely dense media with a very small cost per megabyte of storage, a high data transfer rate, and adequate data compression schemes (preferably lossless) to further reduce the volume. In this case, even thought scientists are reluctant to provide NSSDC with compressed data for distribution, there is little argument against data compression techniques being applied in order to provide for a cost effective backup of the entire digital archive.

As operational mass storage software and hardware systems mature, optical tape would be a prime candidate as an archive backup. The data write rates can be very large (100 MB/s) with a $3,000 cost per reel containing 1 TB of data. Drives are estimated in the range of $200,000 apiece. Another possible media is the digital videotape cassettes which costs about $135 per cassette that can hold data up to 125 gigabytes. Recorders/readers are also in the $200,000 range. Once again, these devices are just now in beta testing with commercial units available within a year and little operational software.

ONLINE INFORMATION AND DATA SYSTEMS

There has been an explosion in the use of available communication technology for the movement and access of mission data and information. Many large universities and nearly every NASA center and other government institutions that work with NASA data have relatively high speed local area networks and many wide area network connections.

There are two major wide area NASA networks that are used extensively; SPAN(2) and the NSN. SPAN contains over 2050 nodes in the United States and is internetworked with over 6000 nodes in the U.S., Europe, Canada, and Japan. Like SPAN, NSN is internetworked with other wide area networks such as ARPANET and the NSFNET that can reach many thousands of computers. In general, these wide area networks are of relatively low speed but are serving a tremendously valuable service for the remote users to gain access to space and Earth science computer resources and to fellow researchers all across the country. Although a modest amount of data is transmitted over the wide area networks, it is not real-time (coming directly from a NASA orbiting spacecraft). The bulk of the wide area traffic is of informational purposes such as remote logon and mail.

The NSSDC is responding to an ever increasing number of user requests by putting more of the data and information about the data in its archive online to electronic access. In this way, the NSSDC can "remain open" past the normal working hours allowing scientists and students to "browse" through the online information looking for an important data set. The online data and information systems that are currently operational at the NSSDC are shown in Table 1.

As will be discussed in the next section, the rapid access to selected data through the NSSDC interactive systems is frequently requested. Since it is not known ahead of time what sections of any one data set will be requested, the NSSDC has loaded all the International Ultraviolet Explorer (IUE) data sets online to accommodate user demand. The data is available through the IUE request system. It is important to note that NSSDC manages its archive offline. Storing all the IUE data online was done with full project co-operation and to gain valuable experience with highly requested online data sets.

IUE ONLINE EXPERIENCE

The NSSDC has loaded all the IUE data (in uncompressed form), consisting of over 61,000 unique star images and spectra, in the NASA Space and Earth Science Computer Center's IBM 3850 Mass Store(3). The Mass Store device is managed by an IBM 3081 system and connected to the NSSDC interactive VAX front ends by a high speed local area network (called SESNET), as shown in Figure 2. An interactive system on the NSSDC VAXs has been created that allows for a remote SPAN user to logon and order IUE data from the electronic Merged Observer Log. Once the exact data segment requested has been identified, the NSSDC request coordinator networks the IUE data from the Mass Store through the local area network and through SPAN to the target computer of the requesting individual. This system became operational in November 1987 and by January 1988 requests were routinely serviced with this system.

In addition, requests for IUE data sent on magnetic tape are easily handled by this system, saving the manual locating of the data. These requests come to the NSSDC from letters, phone calls (not all our users are on computer networks), or electronically.

Figure 3 shows the monthly averages of IUE images requested by individuals (we also send large amounts of IUE data to other archives) from 1979 to 1988. From 1979 to 1987, the only service the NSSDC offered was an offline service resulting in a tape copy of the data being produced and sent to the requestor. The bar graph also shows the monthly number of IUE images in 1988 (averaged over the first four months of that year) requested using the online data requesting and electronic delivery system. As clearly seen, there is a dramatic increase in the amount of IUE data requested in 1988 reaching approximately 350 images/spectra per month. The 1988 requests have come from many scientists at 13 institutions in the United States, Europe, and Canada (locations serviced by SPAN).

Since there has been no new money from NASA Headquarters for increased IUE data analysis, the results clearly show that the tremendous increase in requested data is believed to be due to the convenience this system provides to the user. The following factors are a major part of the user convenience provided by this service:

- Data are loaded to the target system (no tape handling), - Data arrives in the desired format;
- No replacement tape is needed to be sent to the NSSDC (currently the network is a "free" service to the users);
- Rapid turnaround provide the desired data while the scientists are interested.

To be able to use low rate communication networks such as SPAN requires that the size of data requested is relatively small. The IUE example is a good one since the data is managed by the stellar object observed which in itself forms a small enough data subset that it can be easily networked. The amount of time required to network an observation is 2 to 15 minutes depending on the communication rate and the load on the network.

The IUE example serves to illustrate that to better serve some of the users, faster access to requested data is desirable than what the NSSDC has been doing when the data resided offline on magnetic tape.

Many of the most requested data segments come from very large data sets. Keeping large amounts of data online is expensive. Key questions as to the management of the larger volumes of data being archived and promoted to online status must be addressed within the next few years. But if the IUE example is representative of what users need, then to accommodate large amounts of data online, the NSSDC will have to consider use of data compression techniques.

Even though the NSSDC may not compress the data that resides online there are other uses of data compression/decompression techniques when the use of low rate networks are employed to moved the data to a remote location, as discussed in the next example.

NETWORKING OF COMPRESSED AVHRR DATA

Many universities gain access to the Oceanographic data being collected at the University of Miami using SPAN. The University of Miami routinely networks compressed data from the Advanced Very High Resolution Radiometer (AVHRR) instrument onboard a polar orbiting NOAA spacecraft. The AVHRR data is received in real-time are quickly processed (stripping out the infrared portions), compressed, and transmitted to the University of Rhode Island via SPAN where it is decompressed and remapped into a standard set of projections used for several real-time ship activities such as cruise support and chart generation. These images are also networked to Harvard University for their Gulf Stream predication models. In this example, a Lempel-Ziv compression algorithm is used.

SUMMARY

If NASA is going to fly future missions which will produce several orders of magnitude more data than in the past, then it needs to continue to develop better techniques and facilities for data management, data storage and data distribution if it has a chance of preserving and providing the continual extraction of science from its archived data in a timely manner. I predict that as data compression techniques become accepted by the scientific community their implementation at the NSSDC and other data centers will be a necessity. Data compression should be considered an important element of data management for very large data bases.

Access is one of the most important aspects necessary for the proper management of very large data bases. It is projected that the NSSDC will be inundated with data, quadrupling its archive by 1995. Although mass store technology has progressed considerable and must be employed in the managing of very large data bases, it appears that data compression will also play a significant role by providing more data online in conjunction with mass storage capability.

In a similar way, the technology of wide area computer networking has grown considerably. If the speeds of wide area communication networks don't keep up with the demand for the electronic transfer of data, then data compression techniques will also be necessary to implement for data that is transferred over wide area networks.

With respect to the above capabilities, we are somewhat dependent on the ability of the data producers to use the new mass storage technology, the wide spread use of very high speed networks and the acceptance and wide spread use of data compression techniques by the scientific community.

However, the use of lossless data compression techniques as part of the implementation of a complete backup of a very large archive is a viable and, perhaps, necessary step for a cost effective way of preserving irreplaceable digital data archive.

REFERENCES

1) Carper, R., J. Dalton, M. Healey, L. Kempster, J. Martin, F. McCaleb, S. Sobieski, and J. Sos, "Mass Storage Systems for Data Transport in the Early Space Station Era 1992-1998." NASA TM 87826, July 1987.

2) Green, J., V. Thomas, B. Lopez-Swafford, and L, Porter, "Introduction to the Space Physics Analysis Network," NSSDC 87-4, January 1987.

3) Perry, C. M., "The National Space Science Data Center and International Ultraviolet Explorer 1978 - Present," To be Published in the 10th Anniversary IUE Conference Proceedings, March 1988.

ACKNOWLEDGEMENTS

I would like to acknowledge valuable discussions with Charleen Perry who provided updated statistics for the NSSDC IUE requests used in this paper. In addition, I would like to gratefully acknowledge Robert Price who presented this paper at the Data Compression Workshop when I was not able to attend.

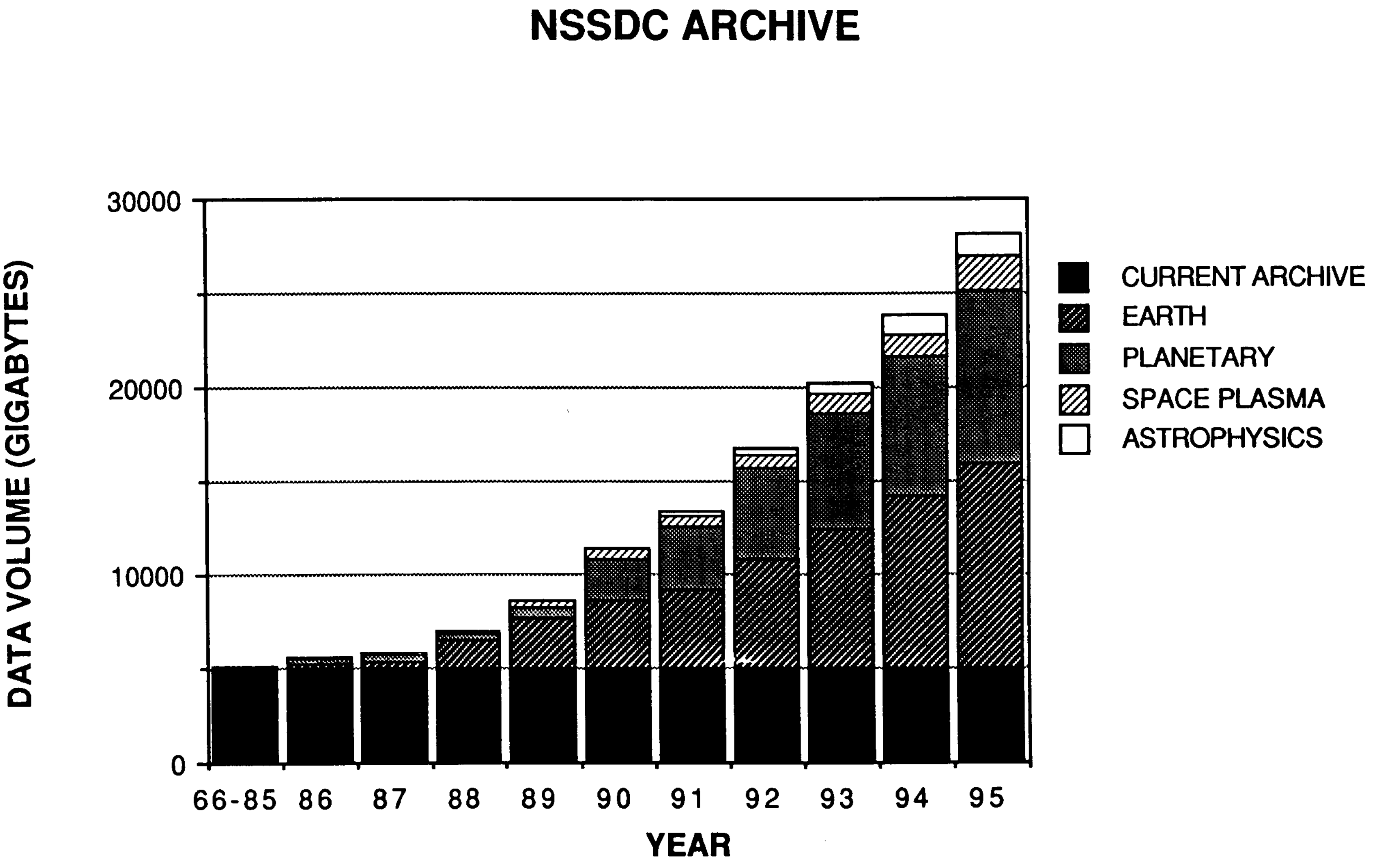

NSSDC ARCHIVE
DATA VOLUME (GIGABYTES)
30000
20000
10000
0
66-85
86
87
88
89
90
91
92
93
94
95
YEAR
■ CURRENT ARCHIVE
▨ EARTH
▩ PLANETARY
▨ SPACE PLASMA
□ ASTROPHYSICS

FIGURE 1

MASS STORAGE DATA ACCESS AT GSFC

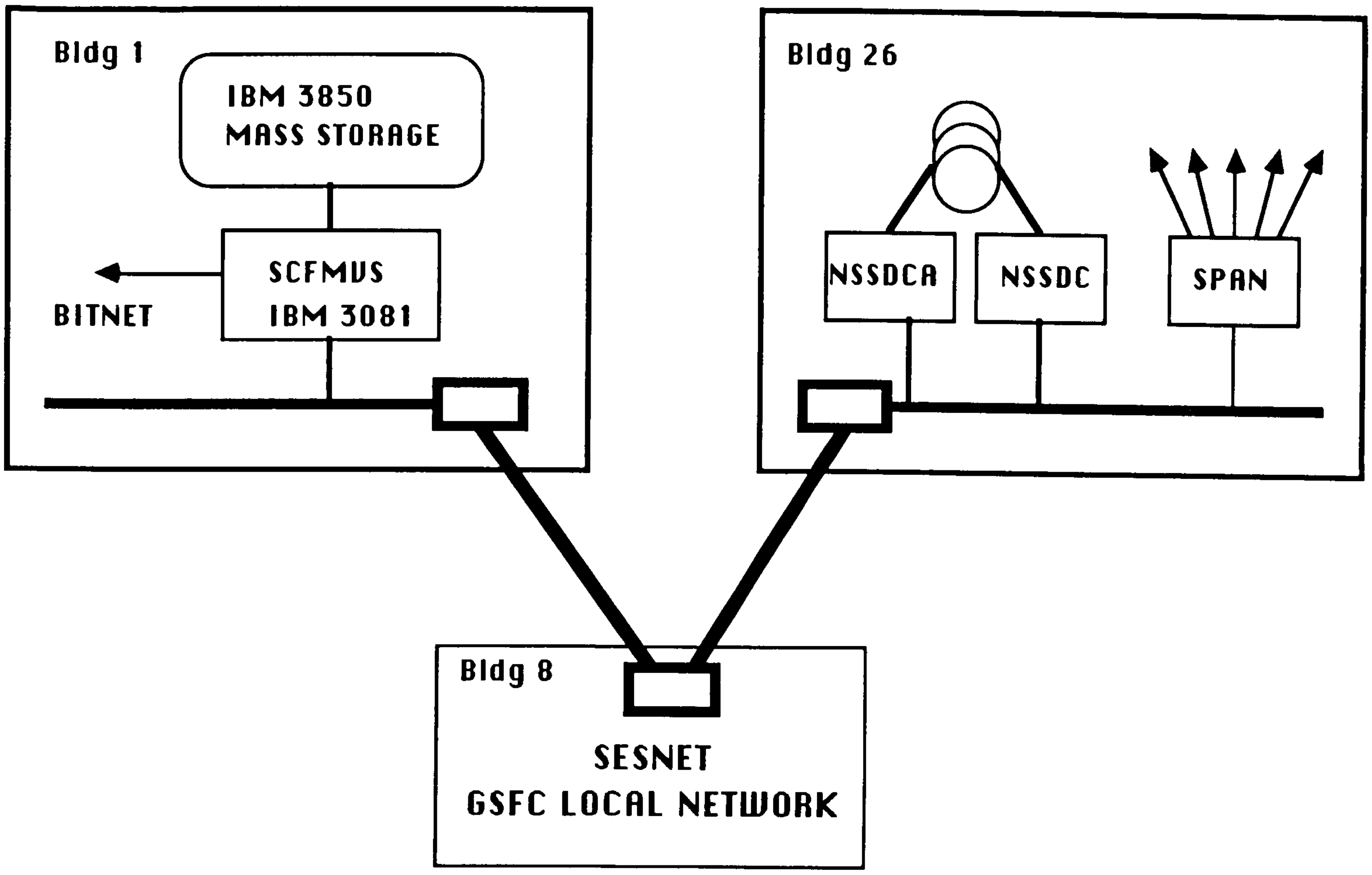


FIGURE 2

IUE IMAGES REQUESTED BY INDIVIDUALS FROM NSSDC ARCHIVE

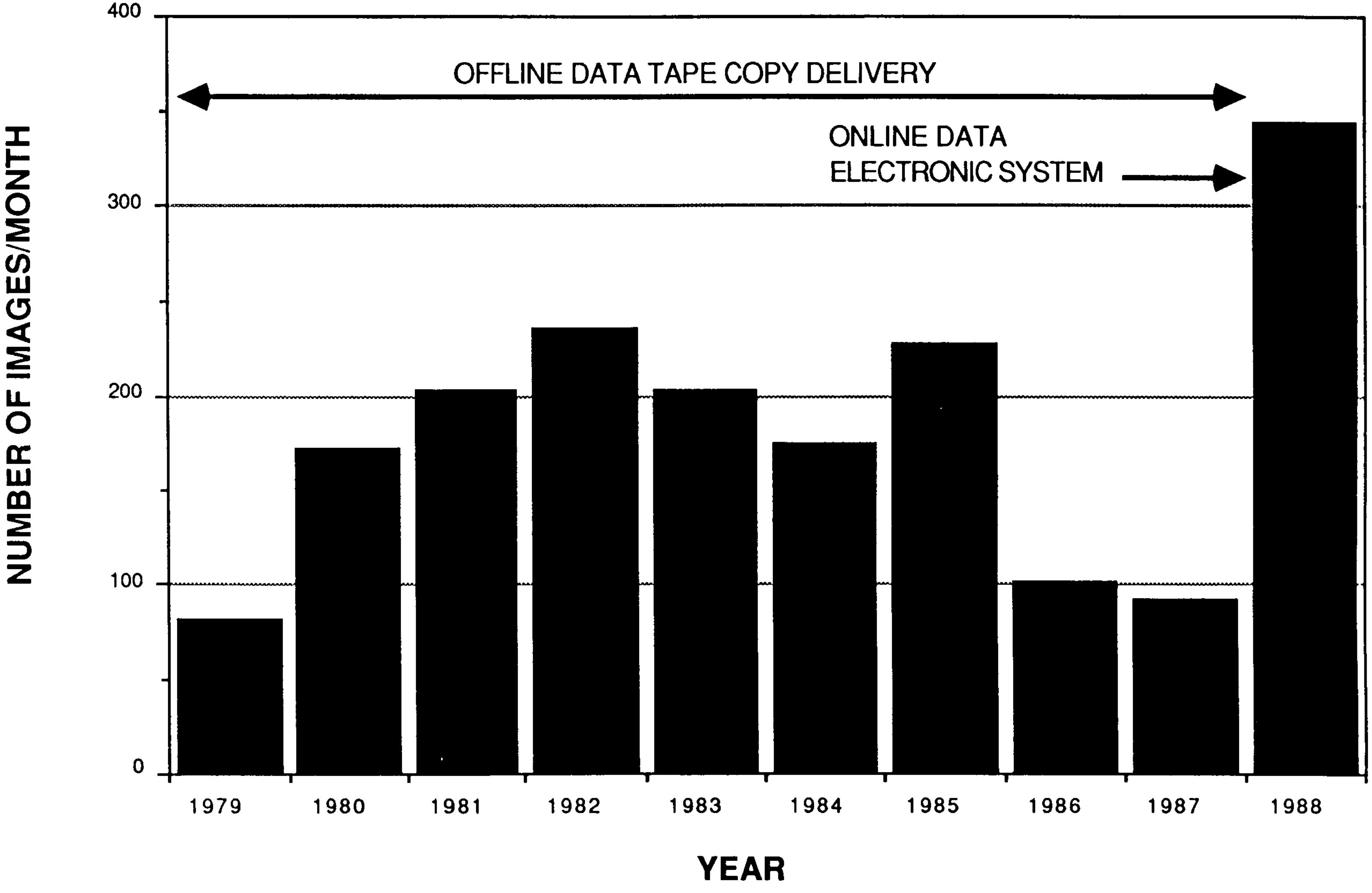


FIGURE 3

TABLE 1
NSSDC ONLINES SYSTEMS

SCIENCE			
DISCIPLINE	SERVICE	INFORMATION	DATA #
All			
	Master Directory	X	
Astrophysics			
	IUE Requests System	X	X *
	ROSAT Info. Manage. Sys.	X	
	Astronomy Catalog Sys.	X	X
	Starcat with SIMBAD acc.	X	X
Atmospheric Science			
	NASA Climate Data System	X	X
	Ozone TOMS data	X	X
Land Sciences			
	Crustal Dynamics	X	X
	Pilot Land Data Systems	X	
Space Plasma Physics			
	Central Online Data Dir.	X	
	Omni Solar Wind Data Sys.	X	X
	Plasma and Field Models	X	X +
	Coordinated Data Ana. Wk.	X	X
General			
	SPAN-Network Info. Center	X	

NOTES:

# Only partial data sets are available
* All available data is online
+ Only software is being distributed

N89-22335

ORBITAL MANEUVERING VEHICLE TELEOPERATION AND VIDEO DATA COMPRESSION

Steve Jones
Marshall Space Flight Center

ABSTRACT

This presentation describes the Orbital Maneuvering Vehicle (OMV) and concepts of teleoperation and video data compression as applied to OMV design and operation.

The OMV provides spacecraft delivery, retrieval, reboost, deboost and viewing services, with ground-control or Space Station operation, through autonomous navigation and pilot controlled maneuvers. The Flight Vehicle (FH) includes a payload/target grapple fixture and a three-point attach mechanism for spacecraft servicing. On-board propulsion is provided by control system (RCS) thrusters, and a cold-gas thruster system. The capability is provided for automatic, on-board attitude control and navigation functions. Communications systems are comprised of S-band RF command, telemetry, and compressed video data links through the TDRSS and GSTDN networks.

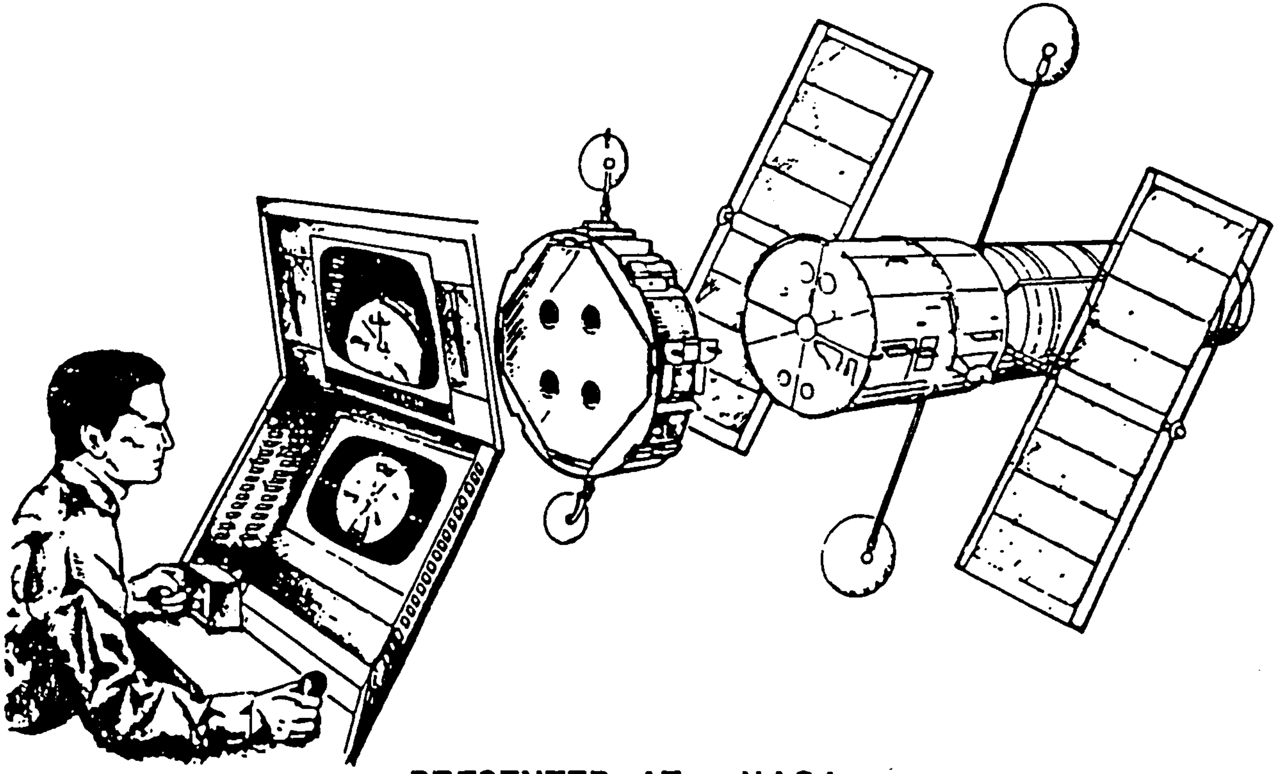
For target viewing and/or docking missions, the OMV navigates autonomously to the vicinity of the target vehicle. The pilot then assumes control of the FV for final maneuvers by observing control-console video monitors and commanding on-board thrusters, through the use of hand controllers at the console. The commands are routed to the FV through TDRSS (or GSTDN), and compressed video (from on-board cameras) is returned through the same network to the pilot. The total round-trip delay time is presently estimated to be 2-3 seconds.

The control console video monitors display a monochrome image at an update rate of five frames per second. Depending upon the mode of operation selected by the pilot, the video resolution is either 255 x

244 pixels, or 510 x 244 pixels. The system compresses the output of one camera into a digital data stream at a rate of 972 kbps (kilobits per second), or can interleave two camera outputs simultaneously into one data stream at 486 kbps per camera.

Since practically all video image redundancy is removed by the compression process, the video reconstruction is particularly sensitive to data transmission bit errors. Concatenated Reed-Solomon and convolution coding are used with helical data interleaving for error detection and correction, and an error-containment process minimizes the propagation of error effects throughout the video image. Video sub-frame replacement (using the appropriate sub-frame in the previous video frame) is used, in the case of a non-correctable error or error burst, to minimize the visual impact to the pilot.

ORBITAL MANEUVERING VEHICLE



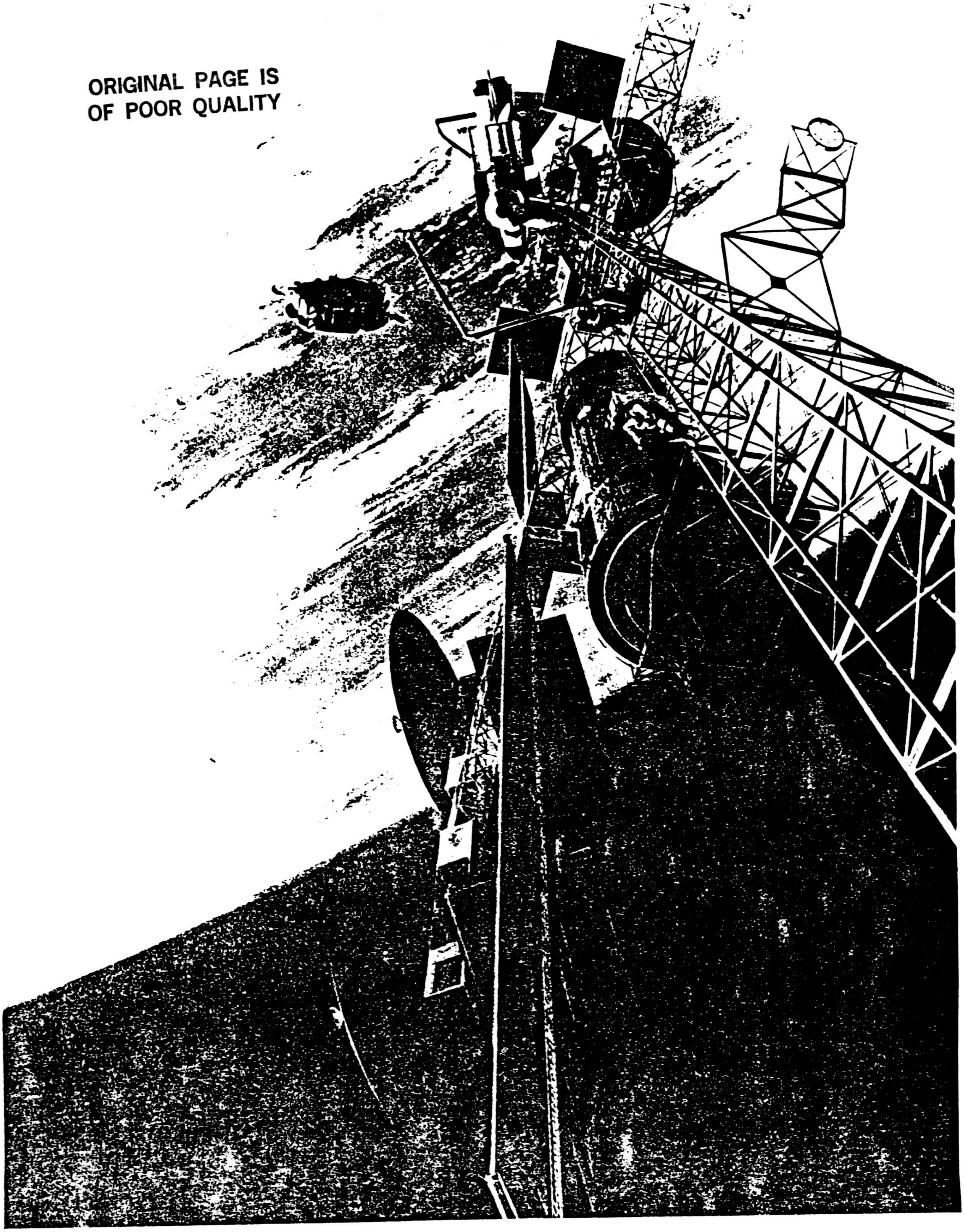
PRESENTED AT:	NASA SCIENTIFIC DATA COMPRESSION WORKSHOP SNOWBIRD, UTAH
BY:	STEVEN R. JONES MSFC/EE63 (205) 544-4373
DATE:	MAY 3, 1988

OMV REQUIREMENTS SUMMARY

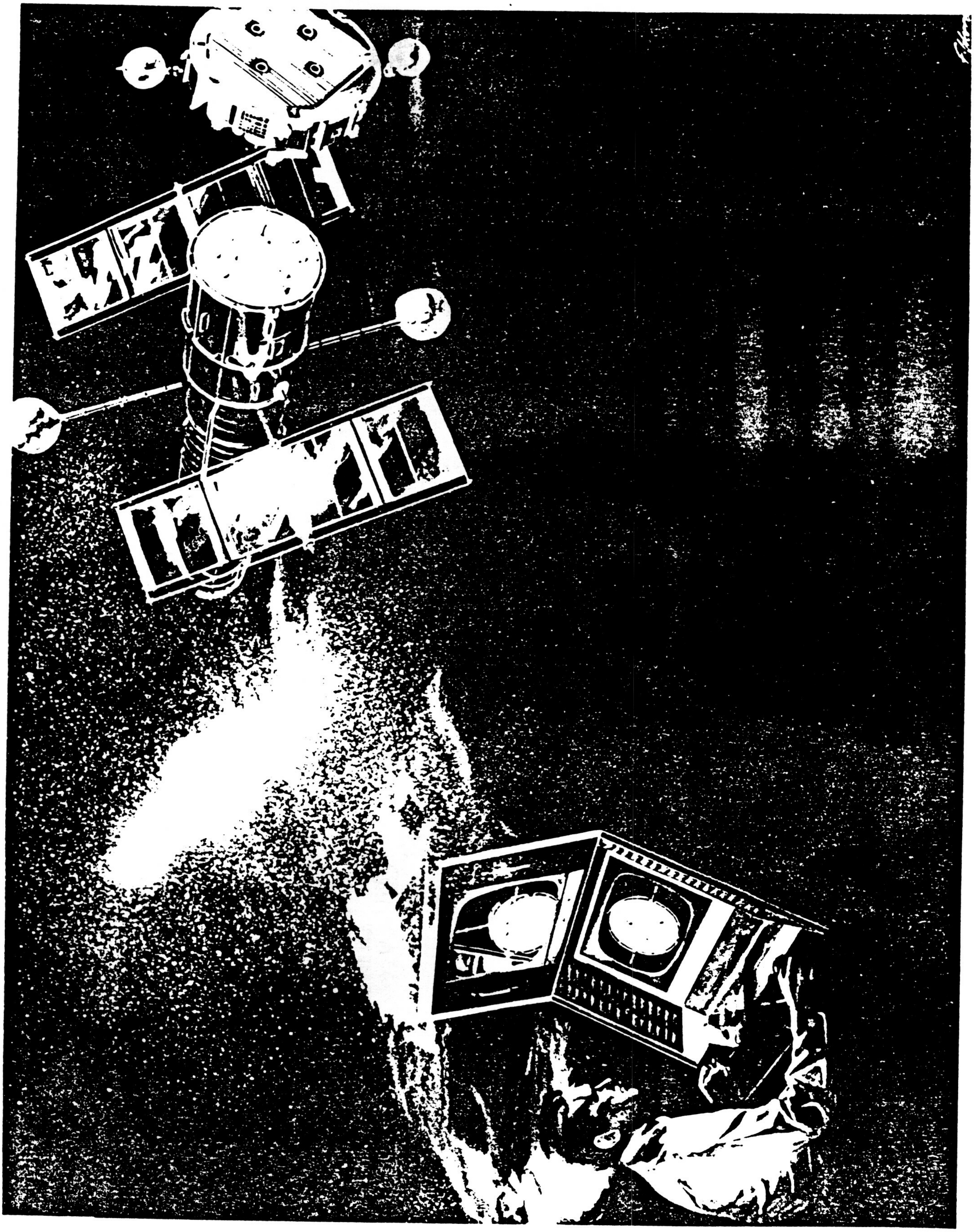
- PROVIDE SPACECRAFT DELIVERY, RETRIEVAL, REBOOST, DEBOOST, AND VIEWING
- SHUTTLE BASED, (ETR & WTR), GROUND CONTROLLED
- SPACE STATION BASED, GROUND AND STATION CONTROL
- AUTOMATIC NAVIGATION & RENDEZVOUS
- MAN IN THE LOOP CONTROL FOR FINAL OPERATIONS
- LOW "G" TRAJECTORY WITH CONTINGENCY RETURNS
- PROVIDE LIMITED RESOURCES TO PAYLOADS
- END EFFECTOR AND 3-PT DOCKING SYSTEMS
- ACCOMMODATE VARIOUS MISSION KITS
- COLD GAS RCS FOR PROXIMITY OPERATIONS
- NINE MONTHS ON-ORBIT (SELF-CONTAINED STORAGE)
- CAPABLE OF ON-ORBIT MAINTENANCE
- TEN YEAR LIFE WITH REFURBISHMENT

ORIGINAL PAGE IS
OF POOR QUALITY

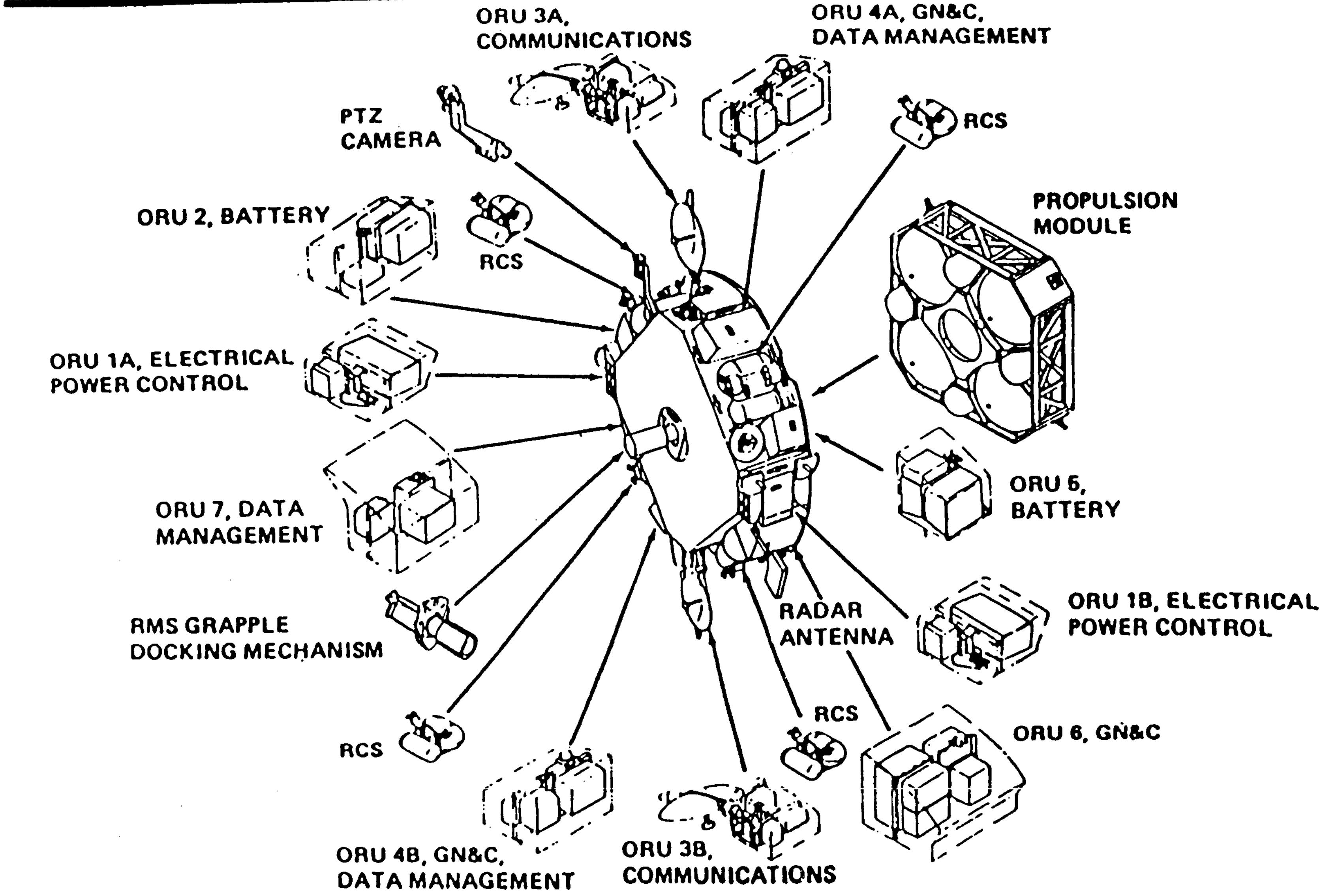
ORIGINAL PAGE IS
OF POOR QUALITY



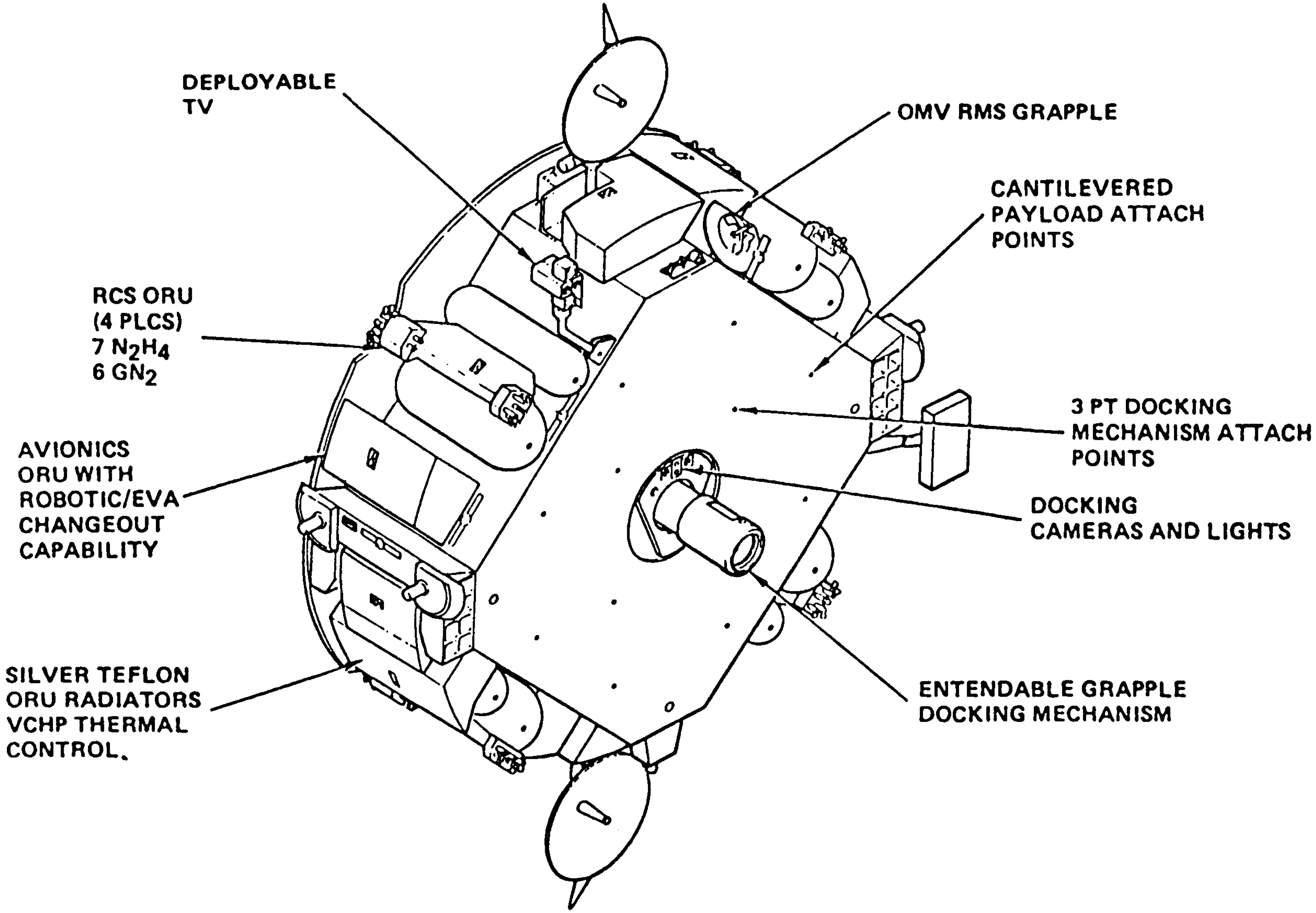
ORIGINAL PAGE IS
OF POOR QUALITY

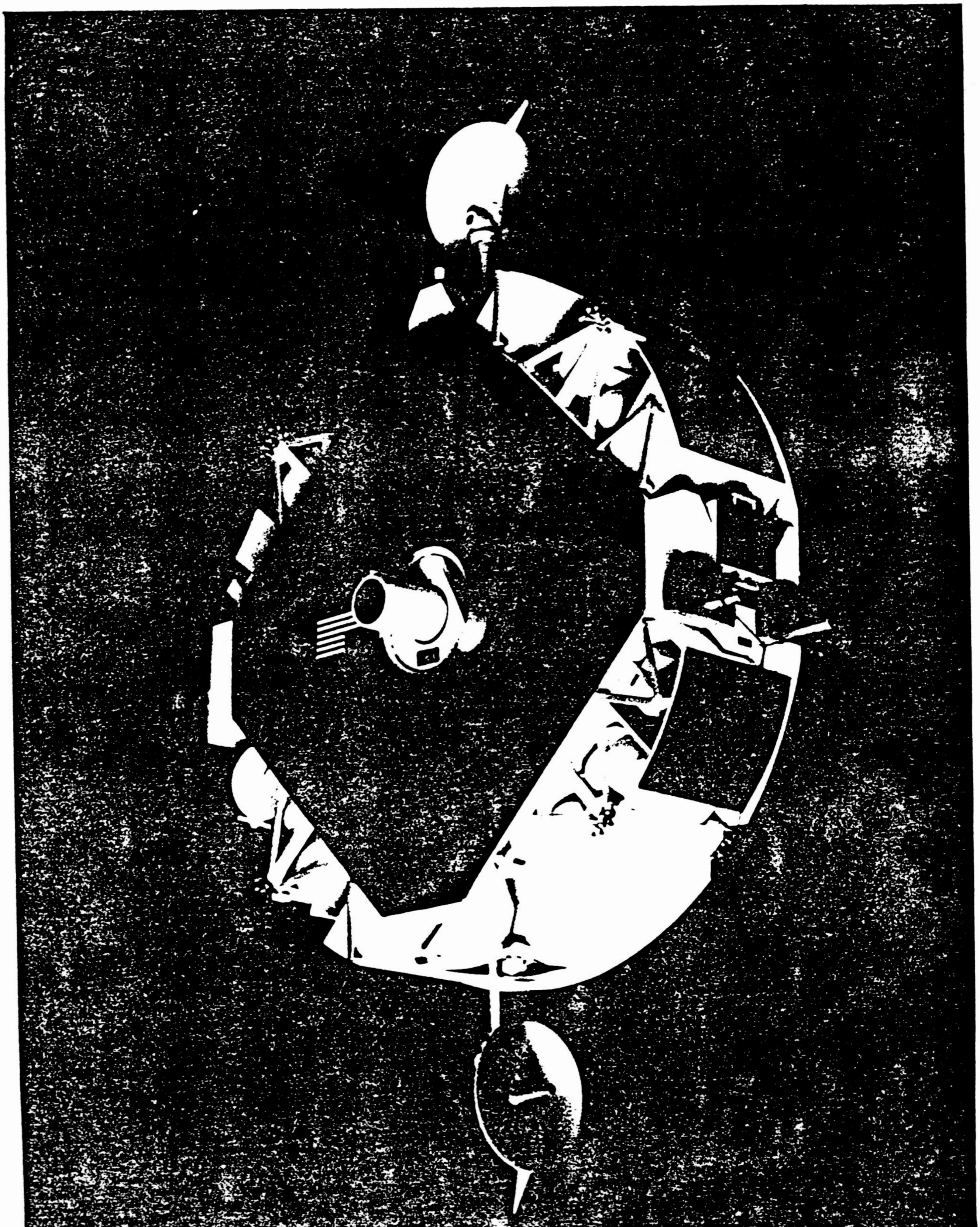


Fully Modular Design and Orbital Replacement Units (ORU's) Enable on Orbit Maintenance

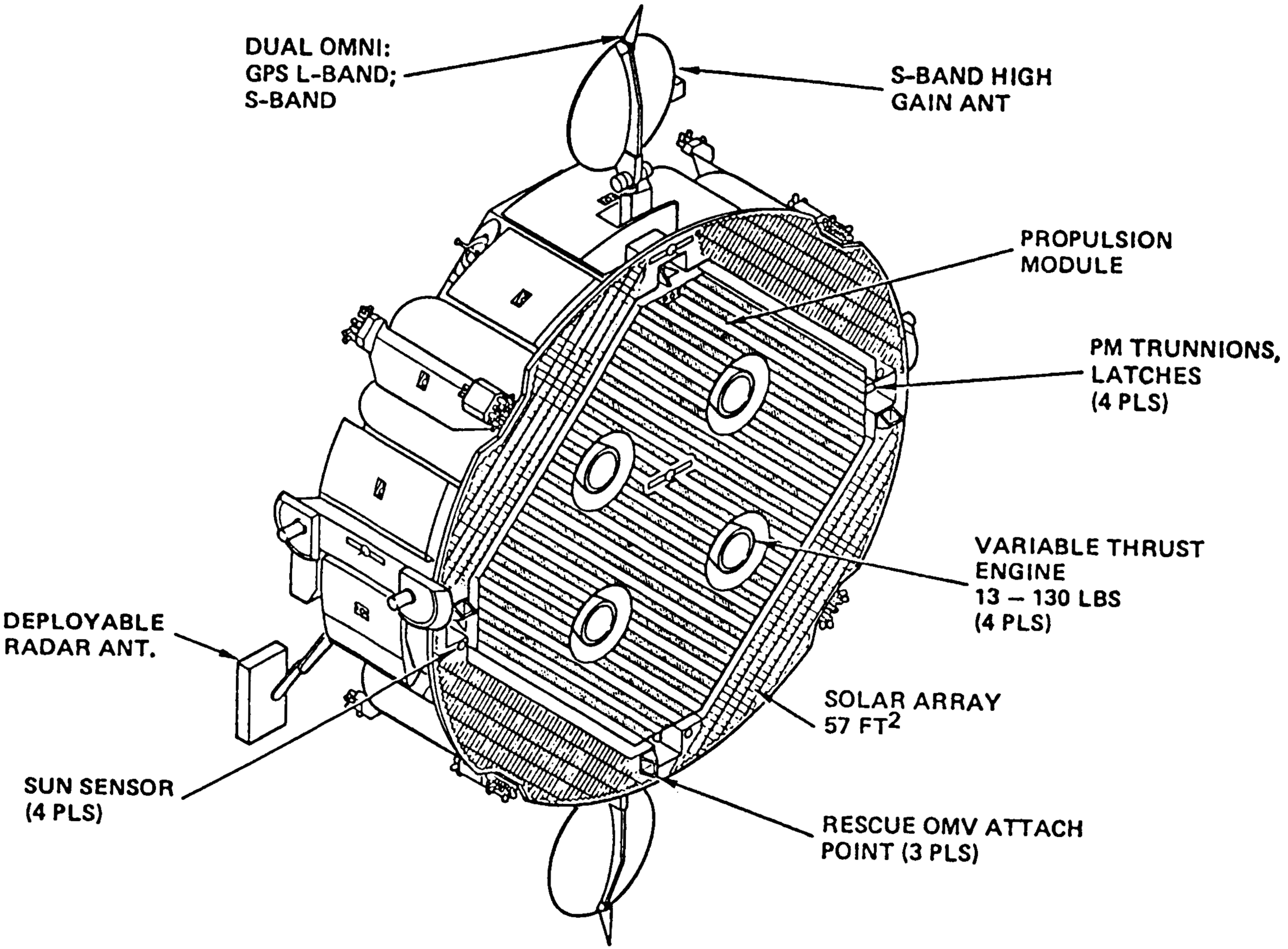


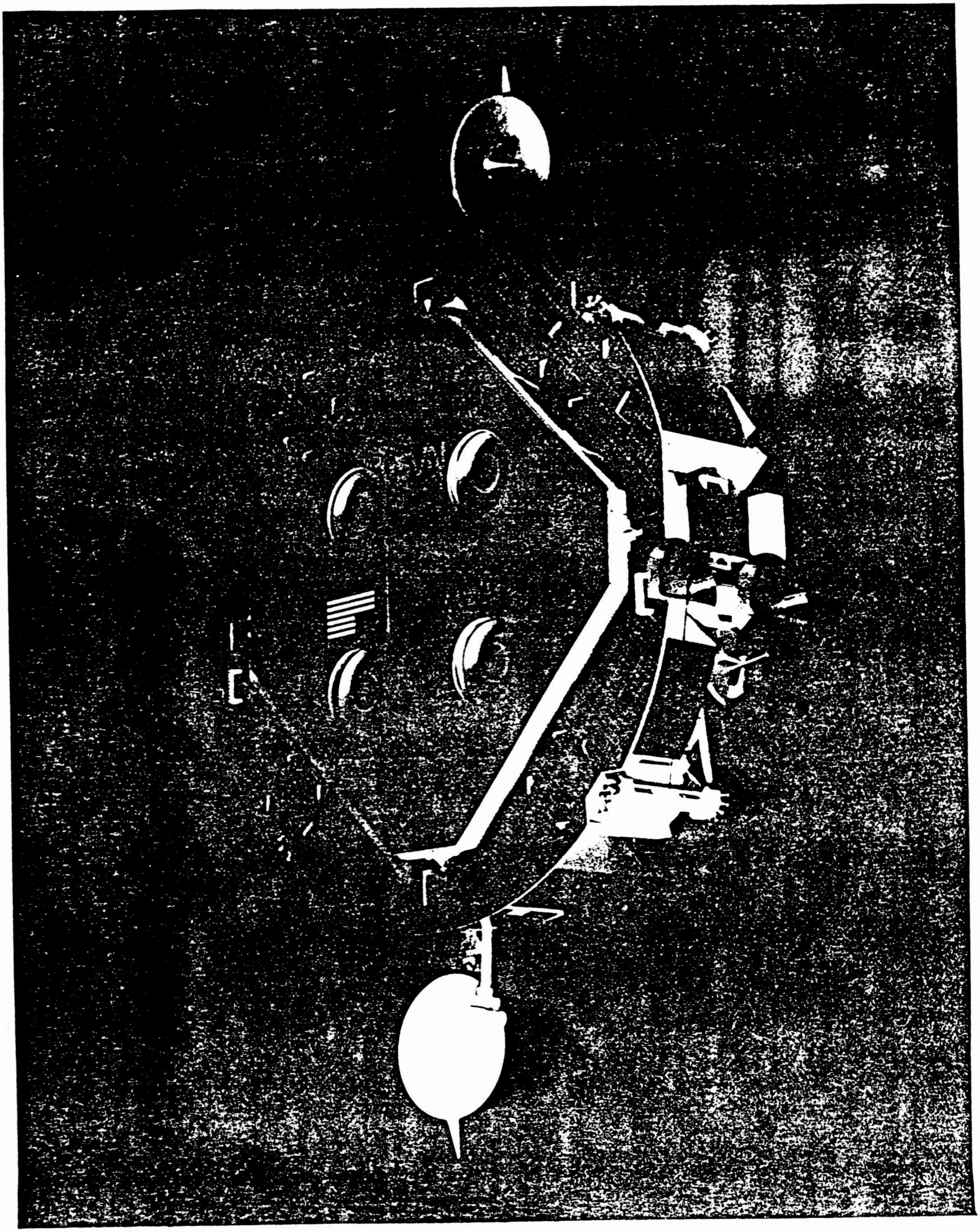
Vehicle – Front Side

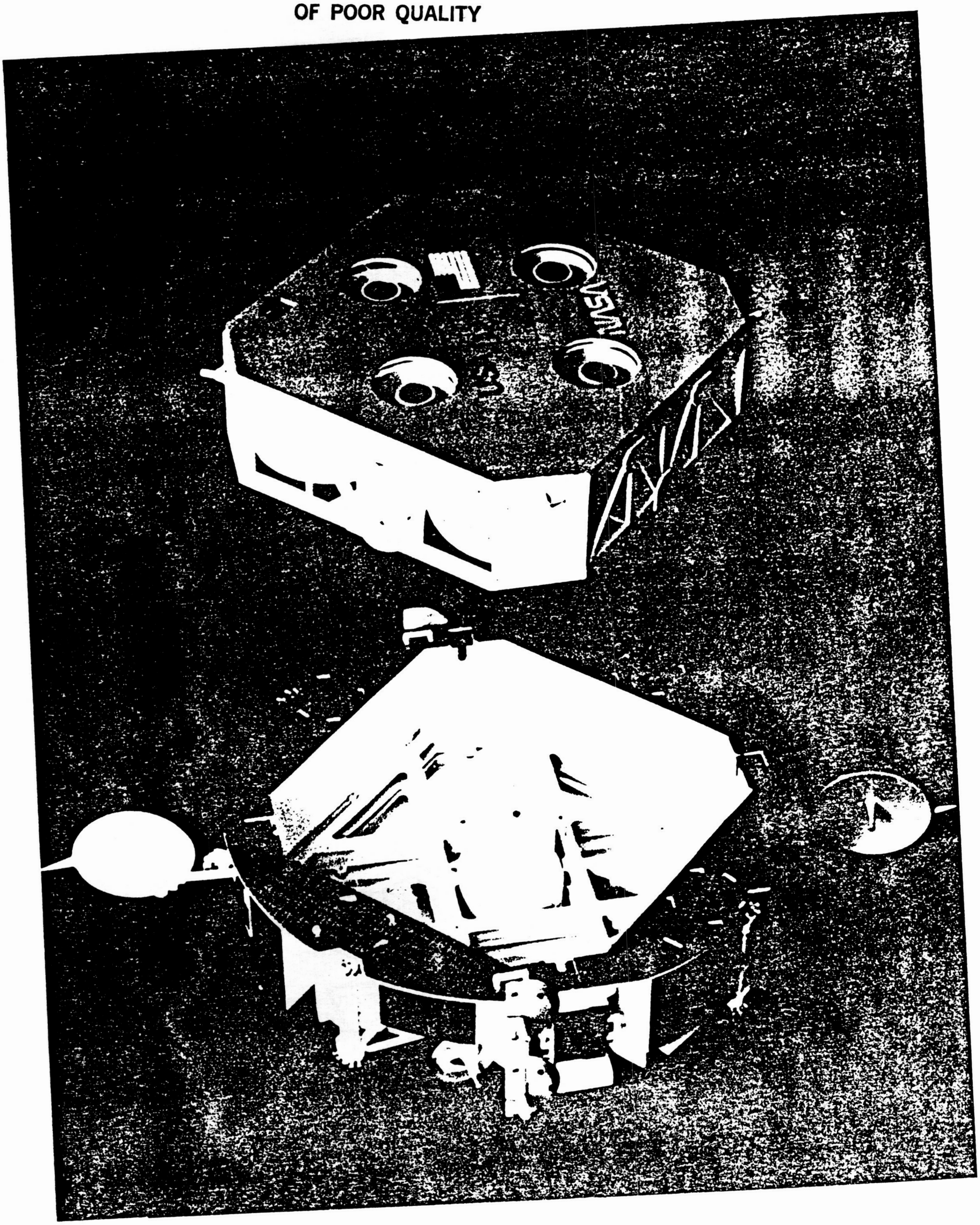

DEPLOYABLE TV
OMV RMS GRAPPLE
CANTILEVERED PAYLOAD ATTACH POINTS
RCS ORU (4 PLCS) 7 N_2H_4 6 GN_2
3 PT DOCKING MECHANISM ATTACH POINTS
AVIONICS ORU WITH ROBOTIC/EVA CHANGEOUT CAPABILITY
DOCKING CAMERAS AND LIGHTS
SILVER TEFLON ORU RADIATORS VCHP THERMAL CONTROL.
ENTENDABLE GRAPPLE DOCKING MECHANISM



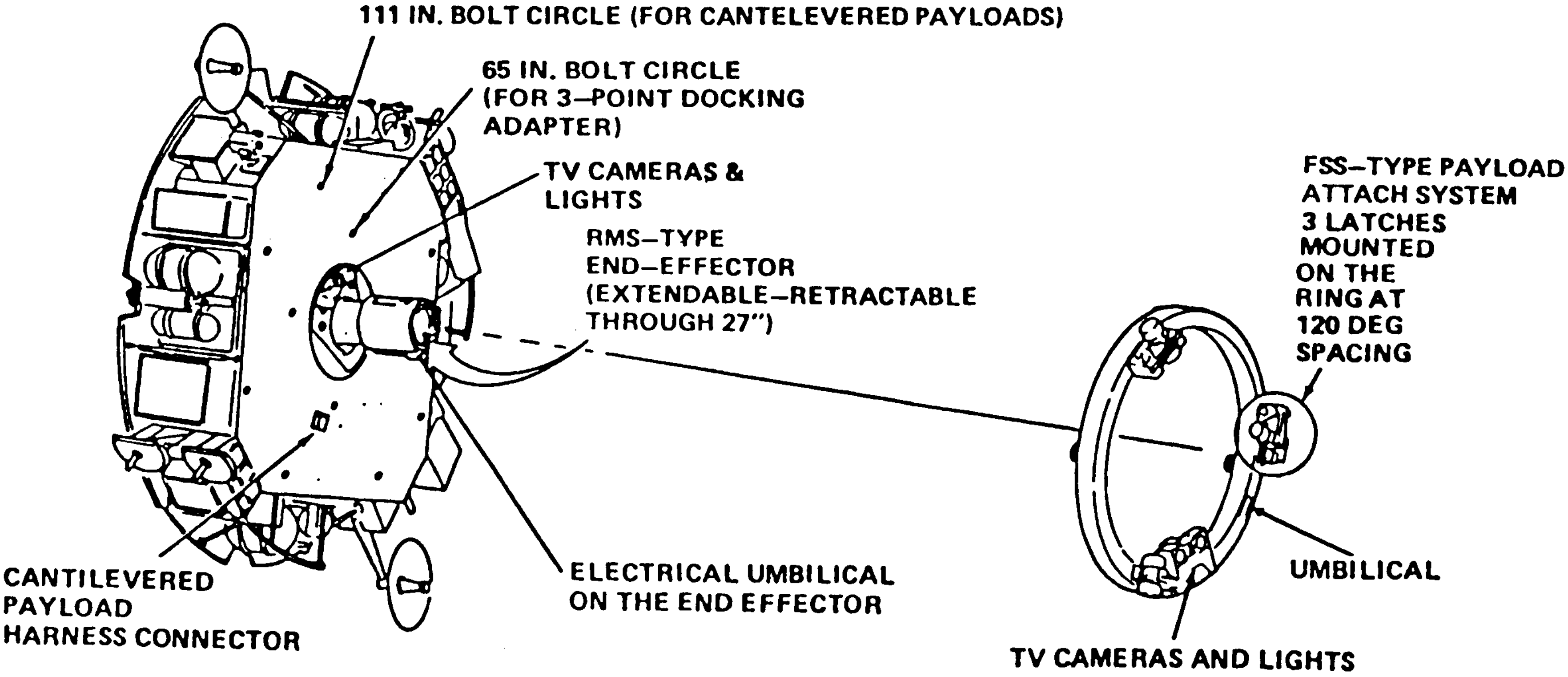
Vehicle – Back Side






NASA

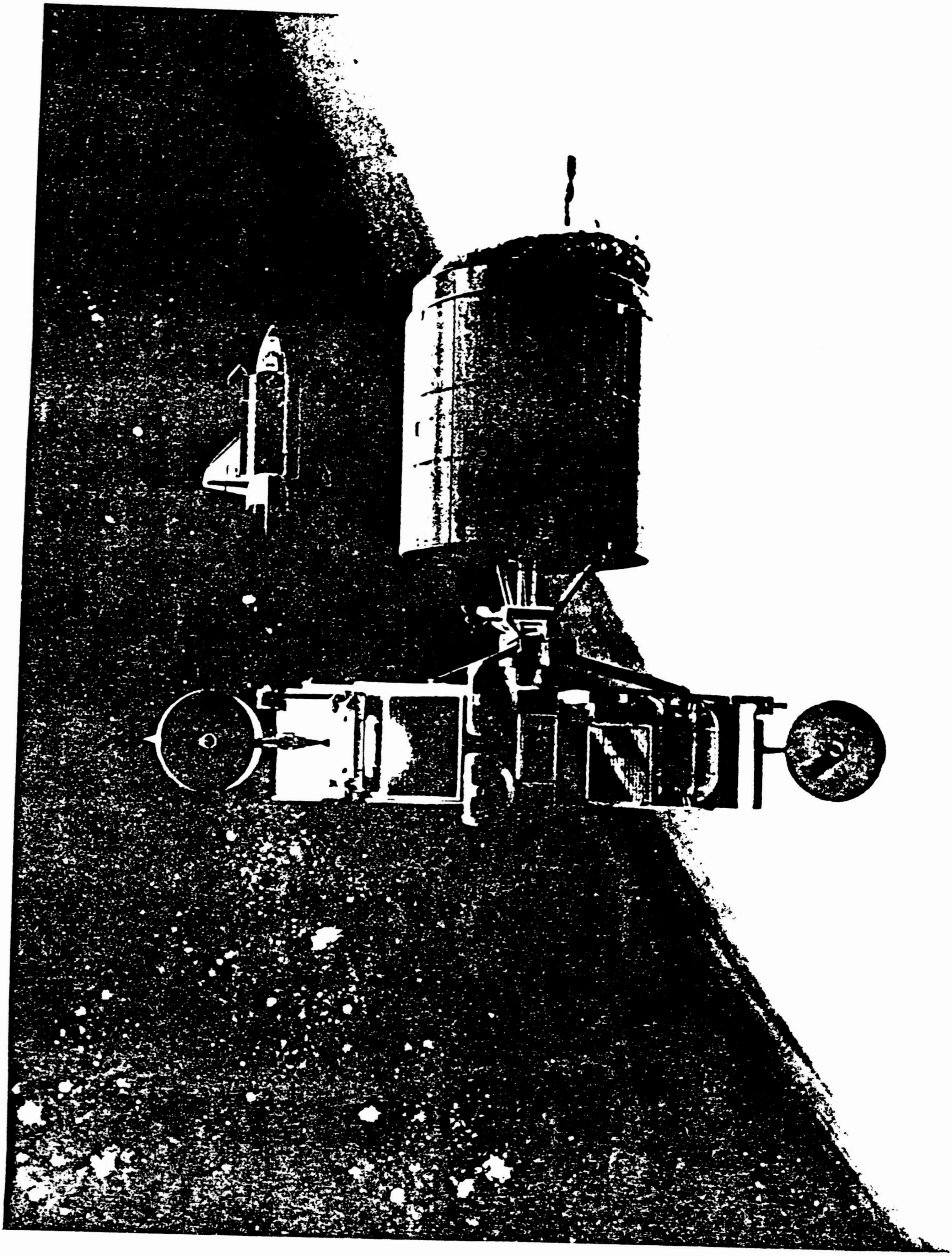
PAYLOAD DOCKING PROVISIONS


111 IN. BOLT CIRCLE (FOR CANTELEVERED PAYLOADS)
65 IN. BOLT CIRCLE (FOR 3–POINT DOCKING ADAPTER)
TV CAMERAS & LIGHTS
RMS–TYPE END–EFFECTOR (EXTENDABLE–RETRACTABLE THROUGH 27")
FSS–TYPE PAYLOAD ATTACH SYSTEM 3 LATCHES MOUNTED ON THE RING AT 120 DEG SPACING
CANTILEVERED PAYLOAD HARNESS CONNECTOR
ELECTRICAL UMBILICAL ON THE END EFFECTOR
UMBILICAL
TV CAMERAS AND LIGHTS

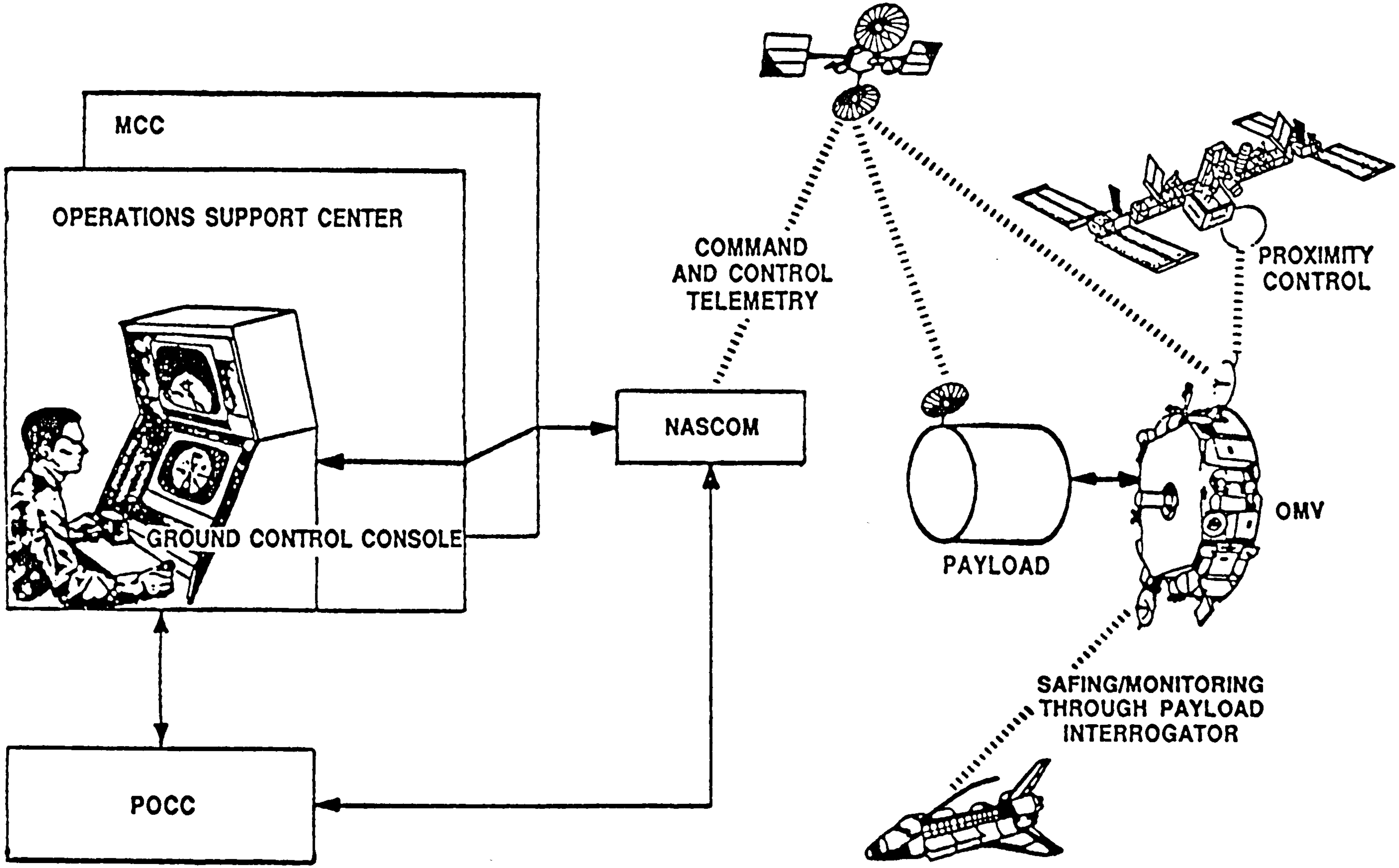
DESIGN CHARACTERISTICS

Width	71 Inches
Diameter	176 Inches (14'8")

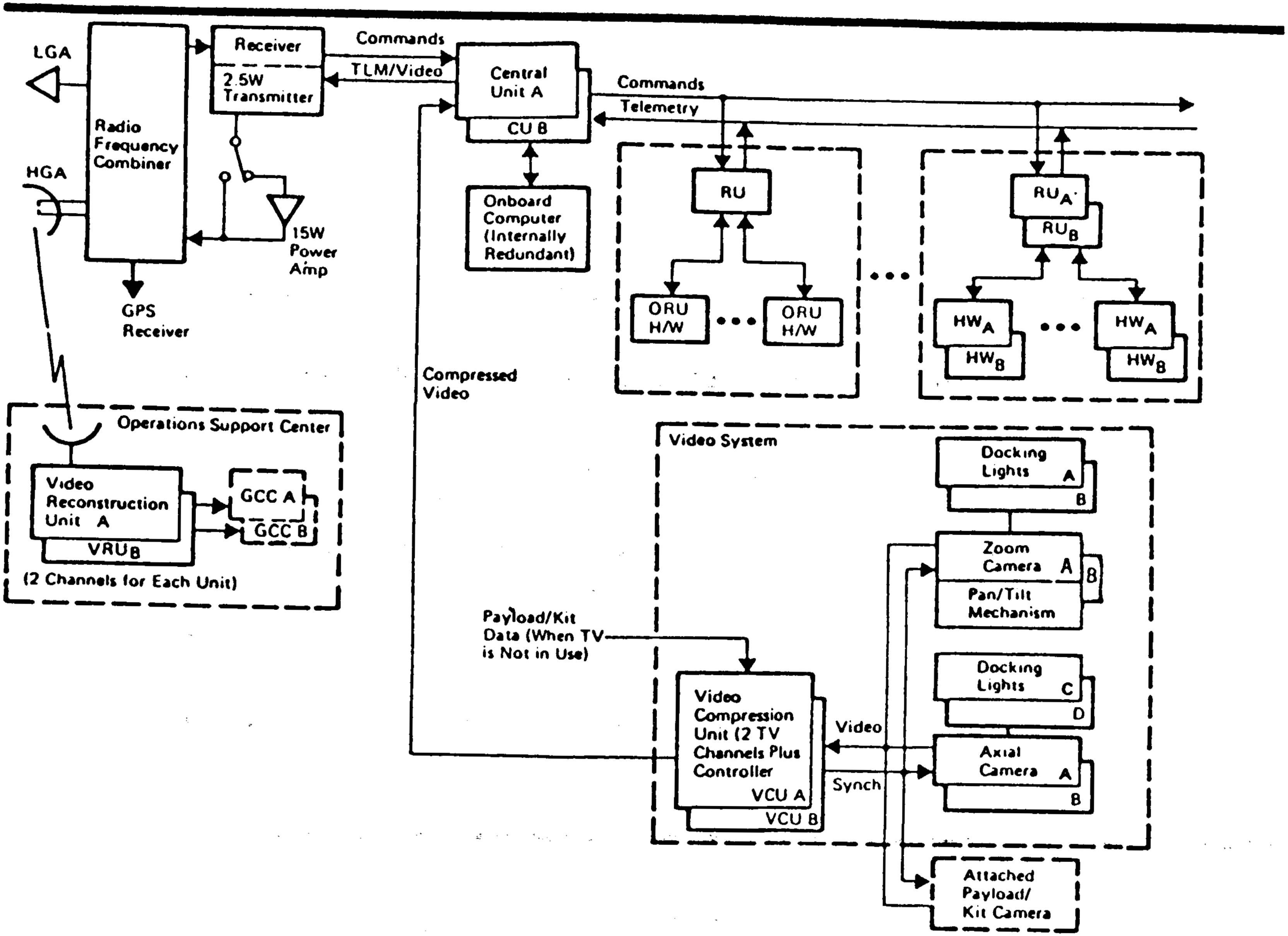
Weight Summary (pounds)	OMV	PM	SRV
Empty (with RGDM)	7340	2125	5215
Burnout (with max. residuals)	7720	2405	5315
Mission Weight			
Minimum	11065	4405	6660
Full	18065	11405	6660
Max. Propellant (Bipropellant/Monopropellant/GN_2)	9000/1180/165	9000	1180/165



OMV OPERATIONAL COMMUNICATIONS


MCC
OPERATIONS SUPPORT CENTER
GROUND CONTROL CONSOLE
POCC
NASCOM
COMMAND AND CONTROL TELEMETRY
PAYLOAD
PROXIMITY CONTROL
OMV
SAFING/MONITORING THROUGH PAYLOAD INTERROGATOR

C&DM SUBSYSTEM



OMV VIDEO COMPRESSION

- COMPRESSED DATA DOWNLINK RATE = 972 KBPS

- MULTIMODE OPERATION

  - TWO CAMERA OUTPUTS INTERLEAVED INTO DATA STREAM
    VIDEO RESOLUTION: 255 X 244 PIXELS (EACH CAMERA)

  - SINGLE CAMERA OUTPUT
    510 X 244 PIXELS VIDEO RESOLUTION

- ON-BOARD TIMING DATA INTERLEAVED INTO DOWNLINK

- RECONSTRUCTED VIDEO IMAGE UPDATED AT 5 FRAMES/SECOND

- ERROR DETECTION/CORRECTION

  - REED - SOLOMON CODING/HELICAL INTERLEAVING

  - CONVOLUTIONAL ENCODING

  - ERROR CONTAINMENT THROUGH VIDEO SUB-FRAME REPLACEMENT

N89-22336

VIDEO REQUIREMENTS FOR MATERIALS PROCESSING EXPERIMENTS IN THE SPACE STATION U.S. LABORATORY

Charles R. Baugher
Space Science Division
Marshall Space Flight Center

ABSTRACT

Full utilization of the potential of the materials research on the Space Station can be achieved only if adequate means are available for interactive experimentation between the science facilities and ground-based investigators. Extensive video interfaces linking these three elements are the only alternative for establishing a viable relation. Because of the limit in the downlink capability, a comprehensive complement of on-board video processing, and video compression. The application of video compression will be an absolute necessity since it's effectiveness will directly impact the quantity of data which will be available to ground investigator teams, and therefore their ability to review the effects of process changes and the experiment progress.

Traditional methodology in materials research investigations has evolved amid close investigator interaction with samples and processes in ground-based laboratories. Effective transition of the research discipline to the space-based laboratories being designed for the Space Station will require a means of continuing this orthodox approach and expanding it to encompass relatively large teams of collaborating scientists at diverse locations. Clearly, the only means of implementing the requirement will be through an efficient and imaginative use of video communications between the Space Station experiments and investigator laboratories. The requirement for comprehensive video presentations of experiment processes and status is further accentuated by the need for on-board crew interaction with experiments which are otherwise extensively contained to avoid the possibility of accidental exposure to noxious processes.

Although it is seemingly a modest requirement, the actual implementation of adequate video monitoring of the Space Station materials processing experiments is complicated by the number of video sources in the planned facility (of the order of ten), their variety, and the several cases in which experiment scientific return will be directly commensurate with the available resolution and/or frame rate. The aggregate bandwidth necessary to freely transfer the raw video information to the ground far exceeds the science allocation form the total 300 Megabits per second being planned for the Station communication link through the Tracking and Data Relay Satellite (TDRS).

To bring the requirement into compliance with the reality of the restrictions, it is apparent that special measures must be implemented within the Station internal video and data management system. An analysis of the science video in terms of a realistic operation model provides guidance on the subsystems necessary to effectively manage the science requirements.

The video from the experiments will range from single frame, high resolution images of the status of crystal growth processes; through requirements involving manipulation which can be readily serviced by the NTSC "media standard"; to a requirement for state-of-the-art in frame rate and resolution to capture the details of certain combustion, fluid flow, and very rapid crystallization phenomena. In general, rapid transmission to the ground is desirable for analysis purposes; however, only the manipulation type of operations will require a near "real-time" type of link. Since the experiment-to-ground delay is not generally of direct importance to the science return, it is possible to make effective use of on-board data buffering and subsequent transmission during periods of opportunity. Indeed, a portion of the video data will be of interest only when analyzed in conjunction with the returned samples and can, therefore, be archived at the Station for return with the regular supply missions.

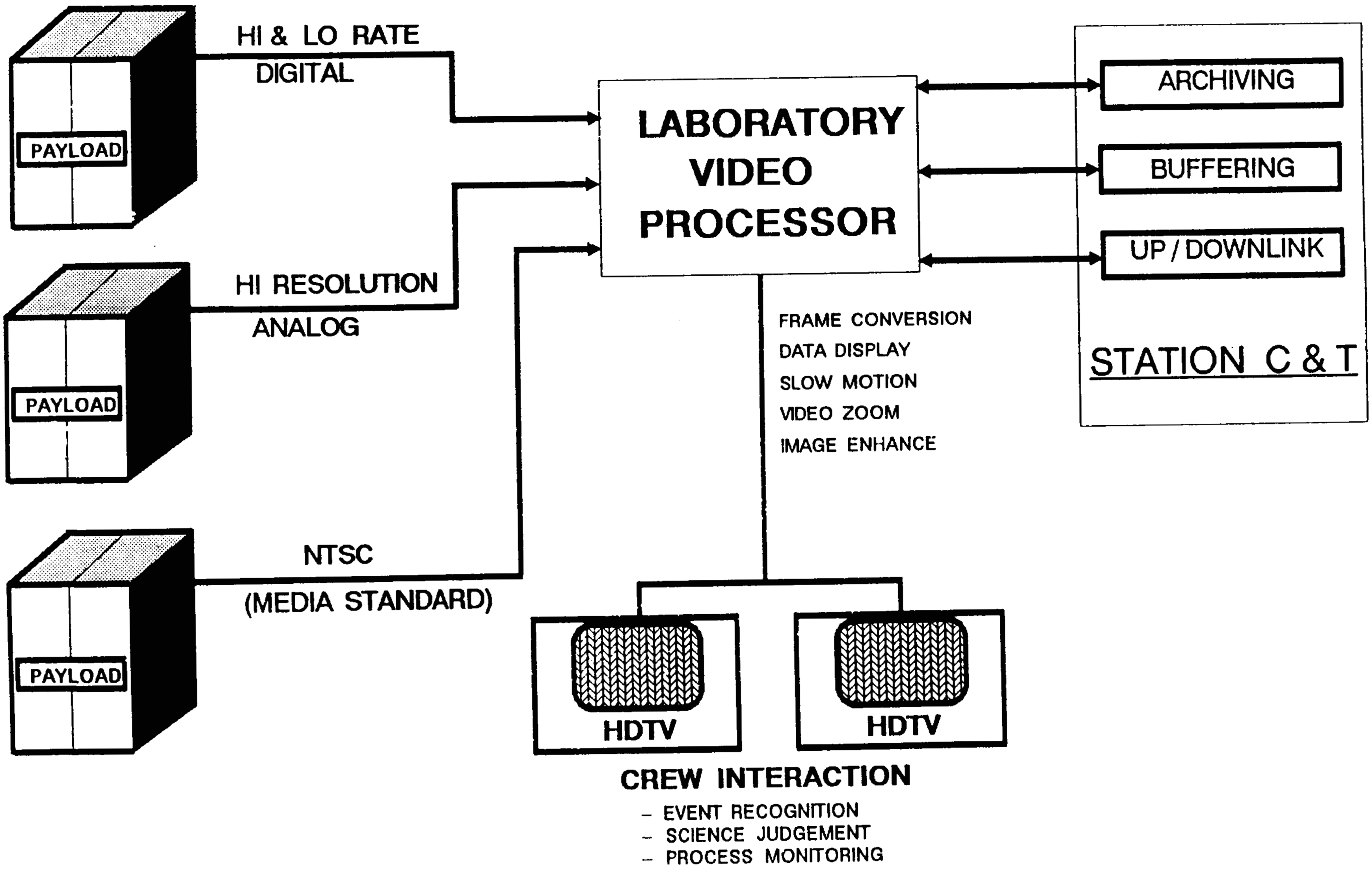
Since the Station will also serve as an active research facility on it own, manned by a scientific crew able to render judgments and synthesize observation, there will be a requirement for on-board interaction with the video data. This interaction will require a set of general purpose video utility tools such as freeze frame, data overlays, and appropriate frame conversion capability. Inclusion of these tools will also alleviate the downlink congestion since scientific crew members can interdict or pre-sort observational data which does not satisfy the experiment requirements. Collectively, the requirements dictate that an experiment dedicated video processing facility will be mandatory as an integral part of the Space Station Laboratory.

However, even with the inclusion of an extensive video processing, recording, and data management capability, adequate operation of the materials processing facilities will likely be limited by the data downlink capability. In some cases this limit will be incurred because adequate buffer requirements are beyond the current state-of-the-art and data will be generated faster than can be reasonably transmitted to storage. In other cases, the limit will occur because experiment reiteration rates will be fixed by the overall available through-put rate to the ground. With regular supply missions between the station and the ground scheduled for 90 day centers, modest generation rates can overwhelm archival storage capability and one or two continuous video sources would saturate the available downlink transmission bandwidth.

Clearly, an aggressive utilization of video compression is indicated for essentially all of the video data from materials experiments, and it is likely that the data will prove responsive to the application of the art. In many of the cases the application will be trivial since the video images will be of interest only where there are changes in slowly varying processes in a small portion of a large field. Manipulative operations occupy the middle ground in the hierarchy of difficulty. In others, such as the details of rapid motion in very low contrast fields, the application will be challenging if it is to prove useful.

In addition to the link between the Station complex will also influence the capability of the overall system to service the video requirements of the science experiments. At the conclusion of the Space Station definition phase these internal rates appeared to be unduly restrictive and tended to establish further restrictions on experiment operations. Should cost considerations preclude a substantial increase in internal data rates, the application of video compression will become an even more critical factor in community. In this case, the burden of compression will fall on individual experiments and will likely be accomplished in a much more severe environment. Compression will be implemented within the restricted space of experiment facilities and will be pushed to the limit of its capability to reduce bandwidth requirements. In this case, it is likely that its full application will only be realized after an evolutionary period involving cooperative research between the two communities of investigators.

OPERATIONS MODEL



U.S. LABORATORY VIDEO PROCESSOR SUBSYSTEM

SUBSYSTEM DEFINITION

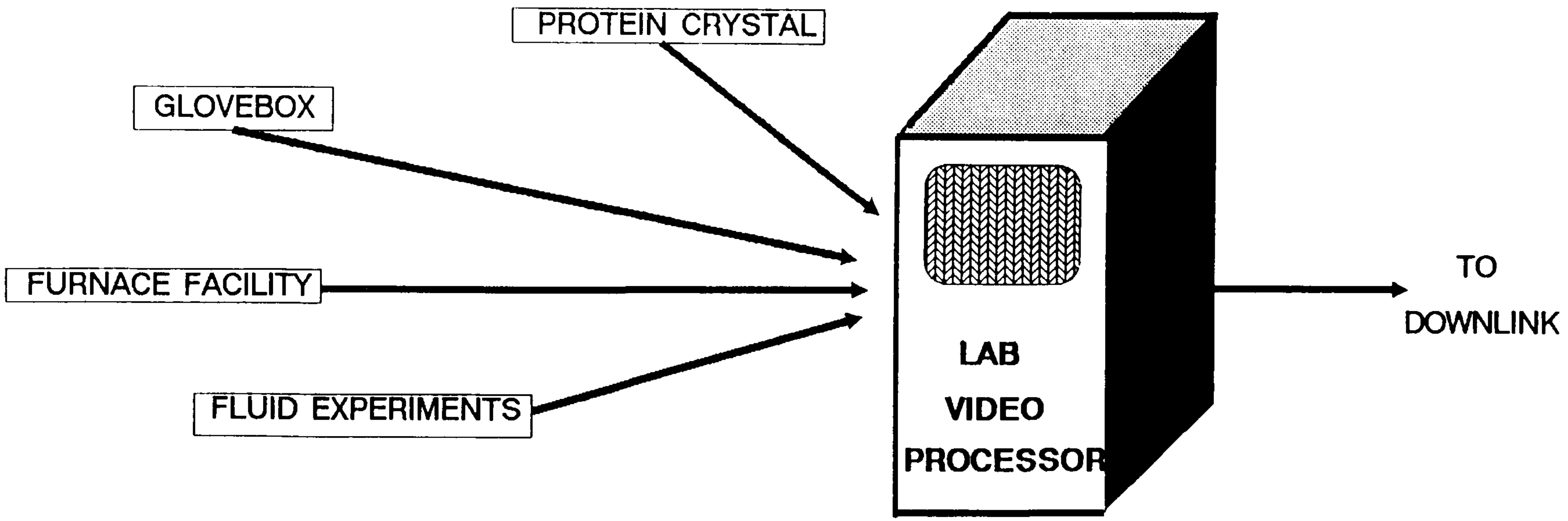
A LABORATORY DEDICATED SUBSYSTEM PROVIDING THE NECESSARY PROCESSING OF VIDEO IMAGES FROM MULTIPLE PAYLOAD FACILITIES IN THE U.S. LABORATORY MODULE TO SUPPORT ON-BOARD SCIENCE OPERATIONS AND EXPERIMENTS AND TO MANAGE AND COORDINATE THE UTILIZATION OF DOWNLINK RESOURCES.

DISPLAY OPERATIONS

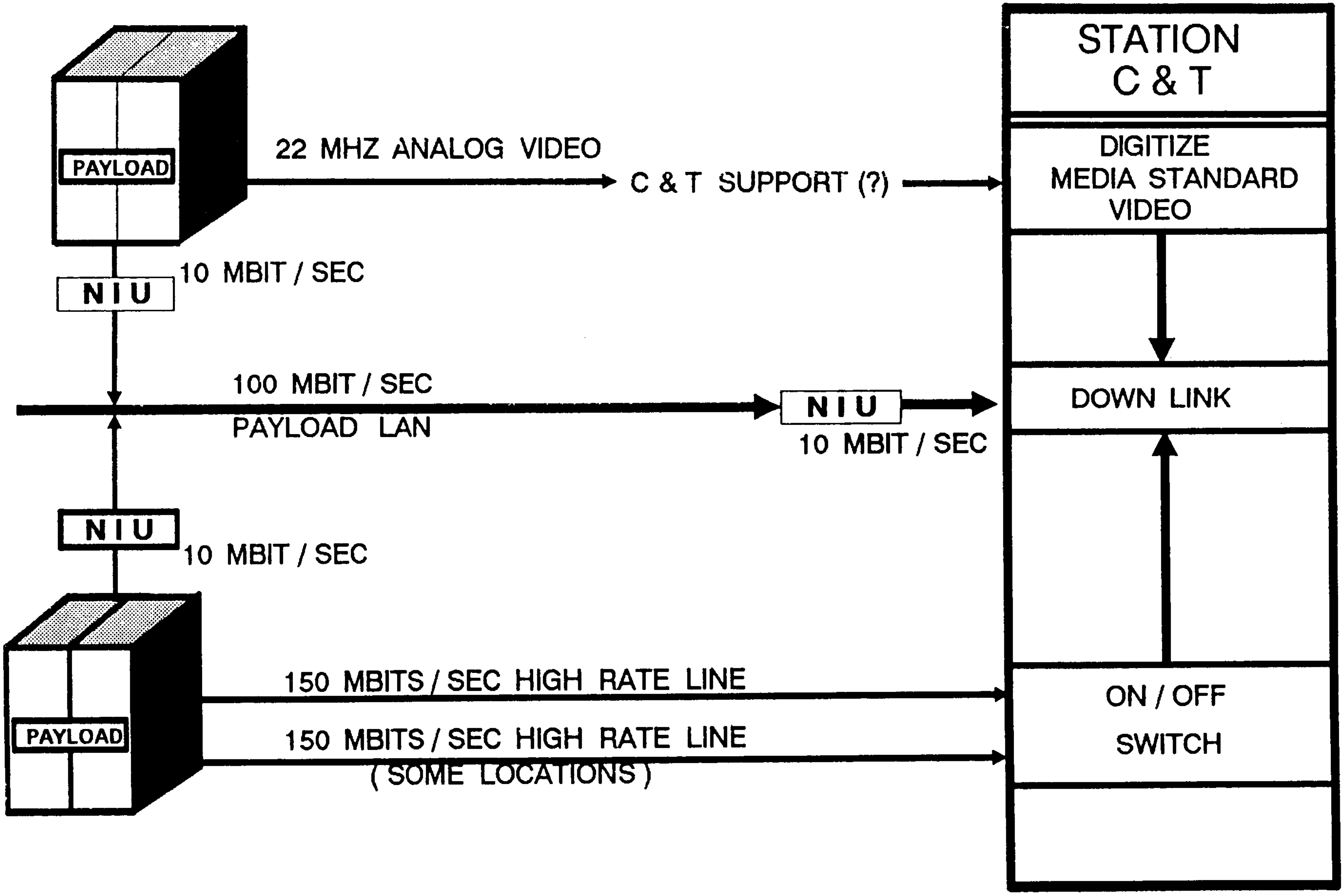
FRAME CONVERSION
IMAGE ENHANCEMENT
UTILITY MANIPULATIONS

TRANSMISSION OPERATIONS

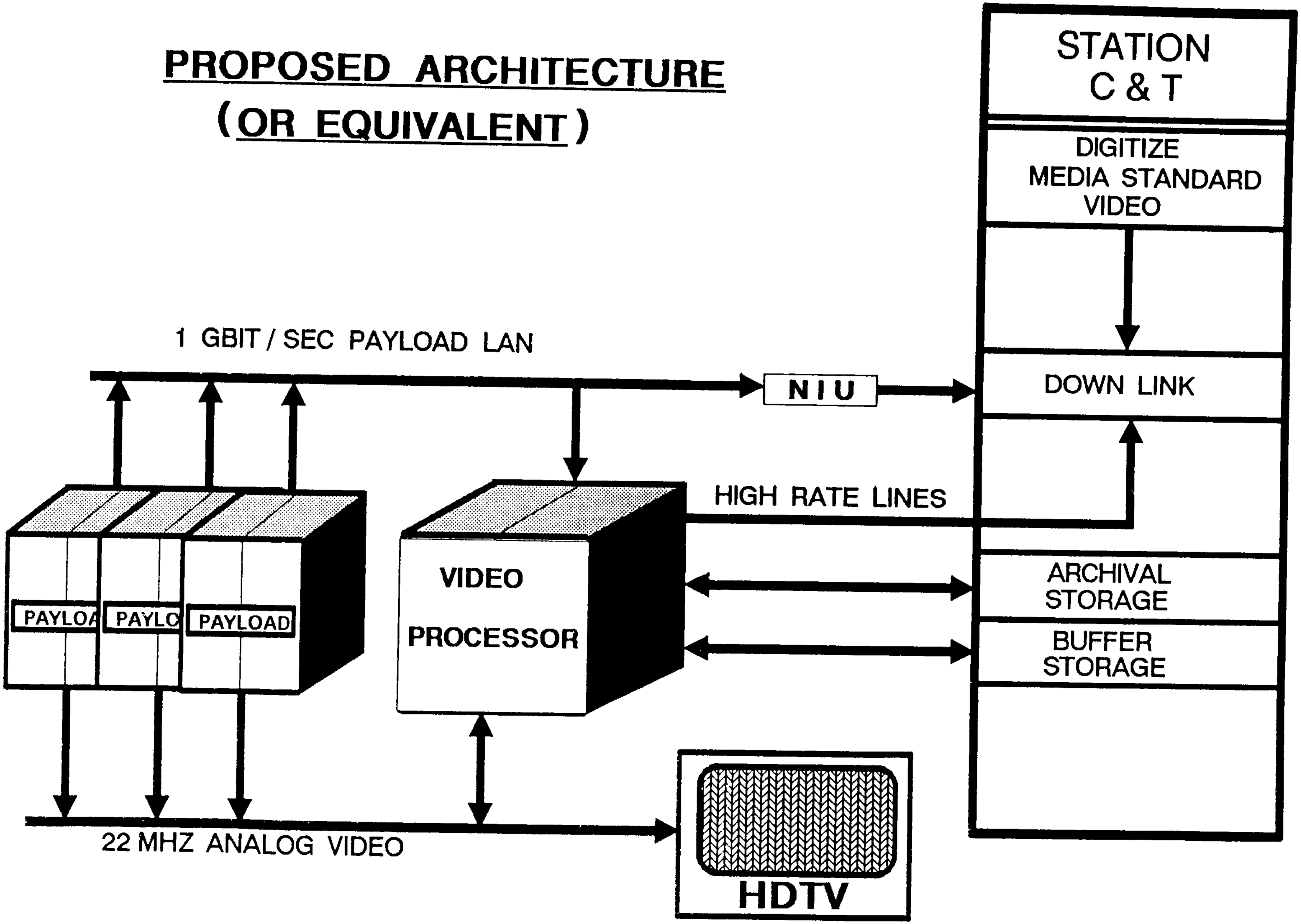
COMPRESSION
MERGING
FORMATTING



CURRENT DMS AND VIDEO ARCHITECTURE



PROPOSED ARCHITECTURE
(OR EQUIVALENT)



COMMUNICATIONS REQUIREMENTS

AND

CONSTRAINTS

ED POSNER

JET PROPULSION LABORATORY

SESSION COORDINATOR

AN OVERVIEW OF REFERENCE USER SERVICES DURING THE ATDRSS ERA

Aaron Weinberg[1]
Stanford Telecommunications, Incorporated

ABSTRACT

The Tracking and Data Relay Satellite System (TDRSS) is an integral part of the overall NASA Space Network (SN) that will continue to evolve into the 1990's. As currently envisioned, the TDRSS space and ground segments will continue supporting the telecommunications and tracking needs of low-earth-orbiting (LEO) user spacecraft until the late 1990's. Projections for the first decade of the 21st century indicate the need for an SN evolution that must accommodate growth in the LEO user population and must further support the introduction of new/improved user services. A central ingredient of this evolution is an Advanced TDRSS (ATDRSS) follow-on to the current TDRSS that must initiate operations by the late 1990's in a manner that permits an orderly transition from the TDRSS to the ATDRSS era. In addition, the ATDRSS must interface with the remainder of the SN elements in a manner that simplifies user access to SN resources, while maximizing user flexibility in satisfying its mission requirements.

NASA is in the process of developing an SN/ATDRSS architectural and operational concept that will satisfy the above goals. To this date, an SN/ATDRSS baseline concept has been established that provides users with an "end-to-end data transport" (ENDAT) service characterized by the following fundamental features:

- A friendly interface with the SN that permits users to obtain services without in-depth knowledge required as to

[1] Supported under contract by NASA/Goddard Space Flight Center

"how ATDRSS works".

- A transition from TDRSS to ATDRSS that is transparent to existing TDRSS users from an operational perspective, but leads to enhanced communications/tracking performance.

- Multiple grades of service that provide users with the flexibility to select an end-to-end service quality (including error-free operation) tailored to the specific mission requirements.

- Growth in the quantity of communication channels, commensurate with the growth in the user population.

- The provision of improved space-to-space RF link efficiency, thereby making ATDRSS support attractive to small users that are currently burdened by the LEO-to-TDRS propagation path.

- The introduction of data rates that exceed 300 Mbps, to permit satisfaction of evolving scientific requirements that may, for example, rely on the availability of digitized high-speed, high-definition TV.

- The application of advanced technologies/techniques that automatically mitigate external phenomena (such as RFI), thereby minimizing service schedule constraints and, hence, maximizing service availability.

Within the context of this baseline, additional service options are currently under investigation that can be readily incorporated with little or no perturbation to the baseline concept. One example is a user capability for autonomous LEO spacecraft navigation. A second example is the introduction of a near-real-time user access feature that potentially alleviates the existing long-lead scheduling process.

On the other hand, potential user services have been identified that are not supportable by the baseline. Most notable here are closure of the zone-of-exclusion (ZOE) and the distribution of data directly from the ATDRS to user premise terminals outside of White Sands. The baseline concept intentionally excludes these features because, to this date, no user requirement has been identified that justifies the associated increase in complexity and cost.

This paper provides an expanded description of the baseline ENDAT concept, from the user perspective, with special emphasis on the TDRSS/ATDRSS evolution. The paper begins with a high-level description of the end-to-end system that identifies the role of ATDRSS; also included is a description of the baseline ATDRSS architecture and its relationship with the TDRSS 1996 baseline. Other key features of the ENDAT service are then expanded upon, including the multiple grades of service, and the RF telecommunications/tracking services to be available. The paper concludes with a description of ATDRSS service options.

AN OVERVIEW OF REFERENCE USER SERVICES DURING THE ATDRSS ERA

19 APRIL 1988

OUTLINE

=======> ● ATDRSS DRIVERS AND OBJECTIVES

- ATDRSS ROLE WITHIN END-TO-END USER SYSTEM
- OVERVIEW OF END-TO-END SERVICES
- ATDRSS SERVICE OPTIONS
- SUMMARY

WHY ATDRSS?

- TDRSS WILL SATISFY SPACE NETWORK (SN) REQUIREMENTS TO THE LATE 1990's

- ADVANCED TDRSS (ATDRSS) MUST ACCOMMODATE GROWTH IN USER-POPULATION/SERVICE-REQUIREMENTS DURING POST-TDRSS ERA

  - ATDRSS INITIATION BY LATE 1990's

  - ORDERLY TDRSS/ATDRSS OPERATIONS CONCEPT EVOLUTION

  - ATDRSS OPERATIONS THROUGH ~2010

USER SERVICE SUPPORT - ACCOMMODATED BY SN/ATDRSS BASELINE

PRINCIPAL DRIVERS

- USER END-TO-END DATA TRANSPORT SERVICE
- SIMPLIFIED/FRIENDLY USER INTERFACE WITH SN
- GROWTH IN QUANTITY OF COMMUNICATION CHANNELS
- MAXIMIZATION OF SERVICE AVAILABILITY
- TDRSS/ATDRSS TRANSITION TRANSPARENCY
- IMPROVED LINK EFFICIENCY
  - ESPECIALLY FOR SMALL, LOW PRIORITY USERS
- DATA RATE EVOLUTION BEYOND 300 MBPS
- RFI-IMPACT MINIMIZATION

OPTIONS

- NEAR-REAL-TIME (DEMAND) USER ACCESS
  - ALLEVIATES LONG-LEAD-TIME SCHEDULING PROCESS
- CONTINUOUS/UNSCHEDULED/AUTONOMOUS NAVIGATION
- $\geq$ 2 USERS IN CLOSE PROXIMITY OPERATIONS
- VIDEO/AUDIO DATA LATENCY, REDUCE TO 600 MS (ROUND TRIP)

ATDRS

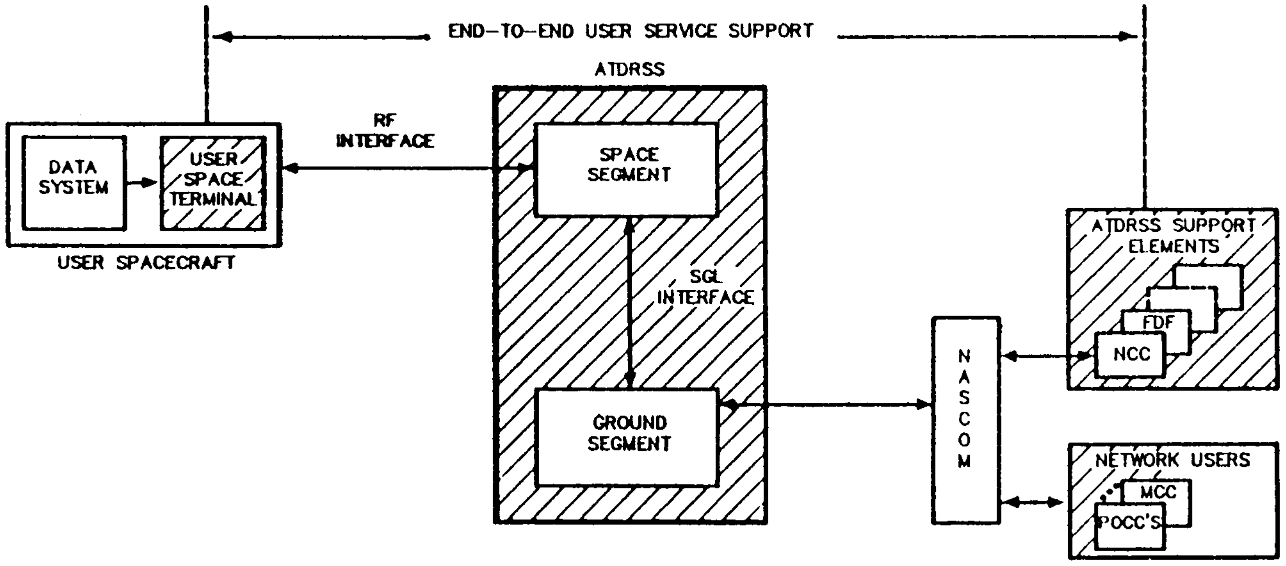
USER SERVICE SUPPORT - NOT ACCOMMODATED BY SN/ATDRSS BASELINE

- SERVICES NOT ACCOMMODATED
  - ZOE CLOSURE
  - DIRECT DATA DISTRIBUTION TO USER PREMISE TERMINALS
- RATIONALE: ABSENCE OF SN OR USER REQUIREMENT
- ABOVE SERVICES ARE OPTIONS THAT ARE SUPPORTABLE VIA ATDRSS BASELINE ENHANCEMENTS
  - ENHANCEMENT, HOWEVER, DEPENDS ON DEMONSTRATION OF NEED

OUTLINE

- ATDRSS DRIVERS AND OBJECTIVES
- ======> ATDRSS ROLE WITHIN END-TO-END USER SYSTEM
- OVERVIEW OF END-TO-END SERVICES
- ATDRSS SERVICE OPTIONS
- SUMMARY

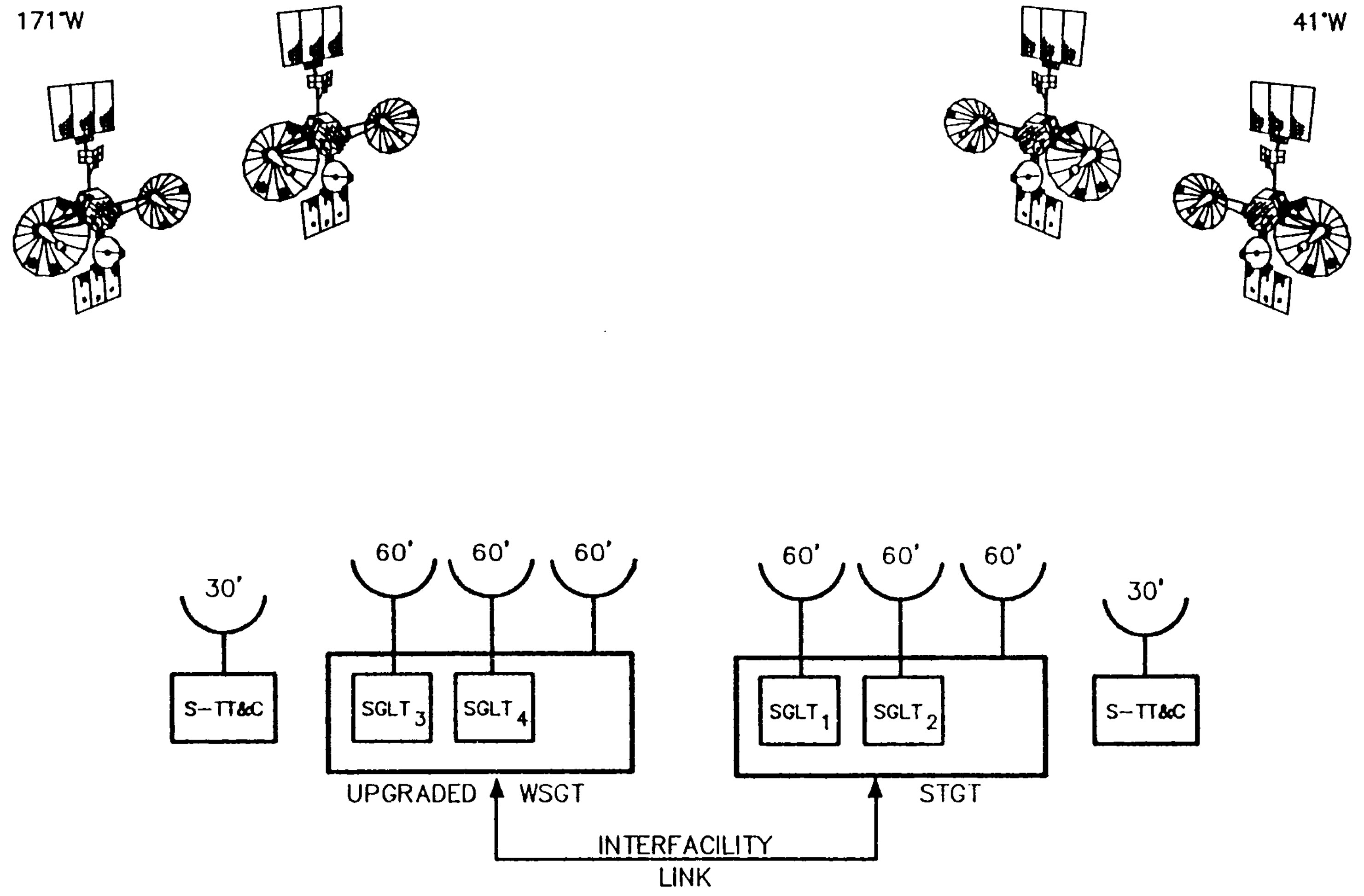
SN/ATDRSS END-TO-END SERVICE CONCEPT



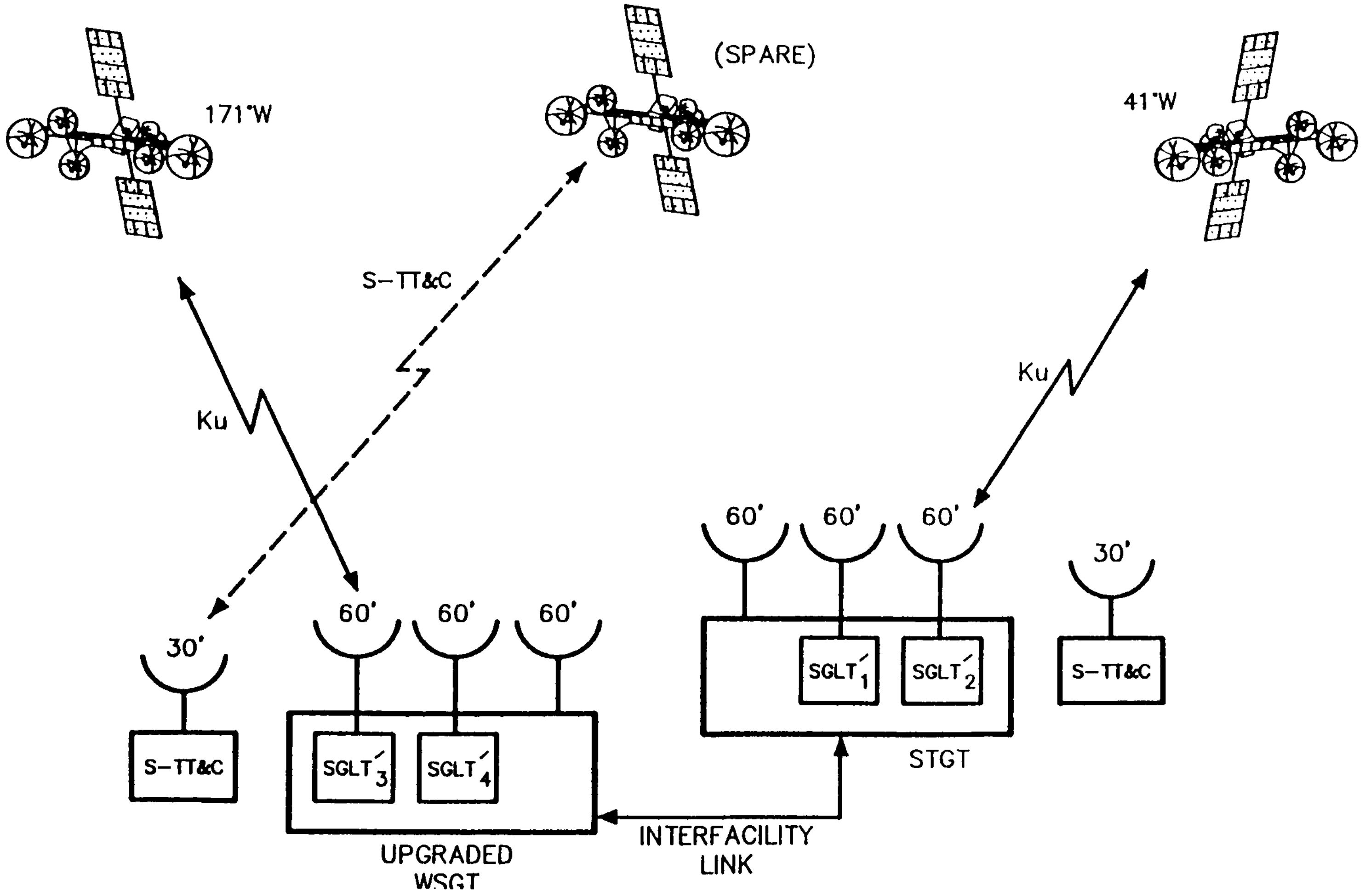
▨ INDICATES THE SCOPE OF END-TO-END DATA TRANSPORT CONCEPT

TDRSS BASELINE ARCHITECTURE

– 1996 POSTURE



REFERENCE ATDRSS ARCHITECTURE



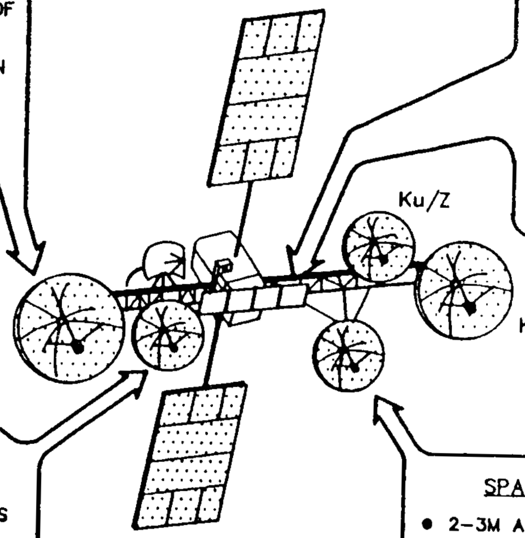
ATDRS

ATDRSS FEATURES

- ENCOMPASSES ALL TDRSS COMMUNICATIONS/TRACKING SERVICES
- INTRODUCES NEW/IMPROVED COMMUNICATIONS/TRACKING SERVICES
- FEWER ATDRSS S/C, LEADS TO SCHEDULING COMPLEXITY REDUCTION
- SUPPORTS SIMULATION AND TESTING
- TDRSS/ATDRSS TRANSITION IS TRANSPARENT FROM "TDRSS-USER" SERVICE PERSPECTIVE
- ATDRSS ARCHITECTURE/OPS CONCEPT SUFFICIENTLY FLEXIBLE TO PERMIT CONTINUED INTRODUCTION OF NEW/IMPROVED SERVICES DURING ATDRSS ERA

ATDRS

ATDRSS SPACECRAFT–REFERENCE FUNCTIONAL CONFIGURATION



DUAL S/Ku SERVICE

- 4.9M ANTENNA
- 1 OF 2
- SUPPORTS CONTINUATION OF EXISTING S/Ku SERVICE
- INCREASED SA LINK MARGIN (ESPECIALLY KSA) VIA STATE-OF-THE-ART TECHNOLOGY

ENHANCED SMA

- 151 ELEMENTS
  - SMA/SSA LINK COMPARABILITY
- ON-BOARD BEAMFORMING
  - 6 CHANNELS, RETURN
  - 2 CHANNELS, FORWARD
- DATA RATES
  - ≤3 MPBS, RETURN
  - ≤10 KBPS, FORWARD
- REDUCED SGL BANDWIDTH
- OPTION: INCLUDES SINGLE BEACON ELEMENT TO SUPPORT
  - AUTONOMOUS USER NAVIGATION
  - NETWORK TIME/FREQUENCY SYNCHRONIZATION
  - ATDRS ORBIT DETERMINATION

DUAL Ku/Z SERVICE

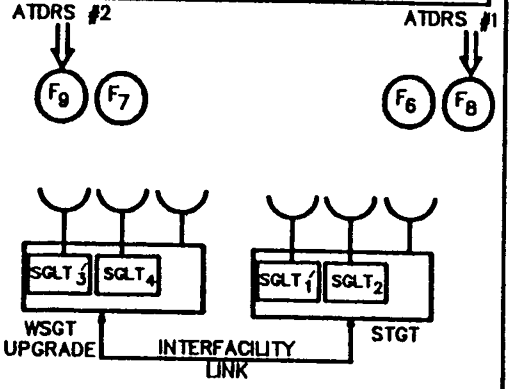
- 3–4.9M ANTENNA
- 1 OF 2
- SUPPORTS TRANSITIONAL KSA USERS WITH LINK QUALITY MEETING OR EXCEEDING TDRS KSA
- ZSA = KaSA OR WSA (TBR)
  - RETURN LINK DATA RATE INCREASE TO 650 MBPS
  - UP TO 9 dB LINK ENHANCEMENT RELATIVE TO TDRSS KSA

SPACE/GROUND LINK (SGL)

- 2–3M ANTENNA
- SINGLE-FEED @ Ku-BAND
- DUAL-POLARIZATION/FREQUENCY-REUSE

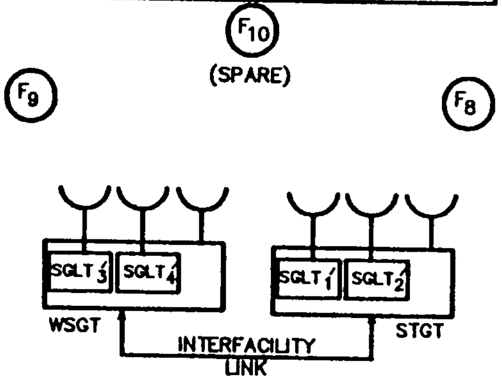
ILLUSTRATIVE TDRSS/ATDRSS EVOLUTION

TRDSS/ATDRSS TRANSITION PHASE −1997 − 1999



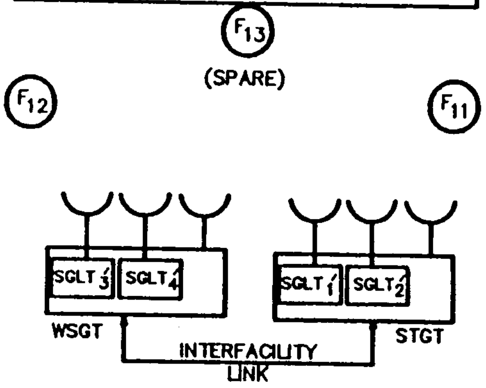
- SUSTAIN 2E/2W CONSTELLATION
- SUSTAIN 6 SA SUPPORT
- FIRST ATDRSS S/C, F8, OPERATIONAL IN 1997 − 1998
- SECOND ATDRSS S/C, F9, OPERATIONAL BY 12/99
- GROUND SEGMENT UPGRADE IN PROGRESS
  - $SGLT_1 \longrightarrow SGLT_1'$ AND $SGLT_3' \longrightarrow SGLT_3'$. COMPLETE BY 12/99

EARLY ATDRSS ERA 2000 − 2003



- TRANSITION TO ATRSS SERVICE REQUIREMENTS, 1/2000
- LAUNCH F_{10} BY 2003 TO COMPLETE E/C/W CONSTELLATION
- RETIRE REMAINING TDRSS S/C ONCE ATDRSS E/C/W CONSTELLATION COMPLETE
- COMPLETE GROUND SEGMENT UPGRADE
  - $SGLT_2 \longrightarrow SGLT_2'$
  - $SGLT_4 \longrightarrow SGLT_4'$

REMAINDER OF ATDRSS ERA −2004 − 2012



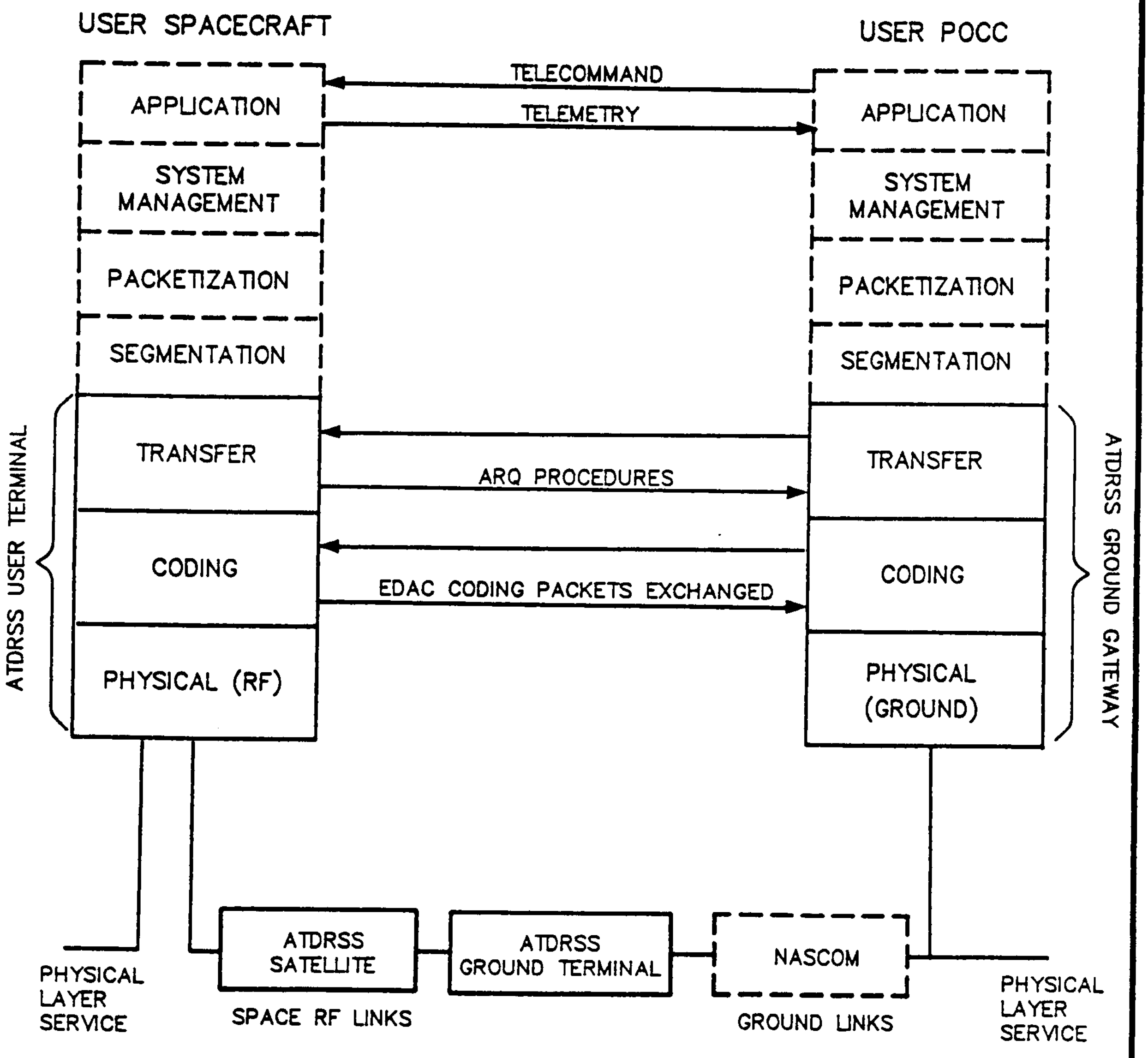
- REPLENISH S/C AS NECESSARY
- SUSTAIN 3 S/C CONSTELLATION
  - CONSTELLATION ADEQUATE TO SUPPORT POTENTIAL GROWTH IN SERVICE REQUIREMENTS BY 2010

ATDRS

OUTLINE

- ATDRSS DRIVERS AND OBJECTIVES
- ATDRSS ROLE WITHIN END-TO-END USER SYSTEM
- =======> OVERVIEW OF END-TO-END SERVICES
- ATDRSS SERVICE OPTIONS
- SUMMARY

ATDRSS LOW DATA RATE SERVICE MODEL



USER SPACECRAFT

USER POCC

APPLICATION

SYSTEM MANAGEMENT

PACKETIZATION

SEGMENTATION

TRANSFER

CODING

PHYSICAL (RF)

APPLICATION

SYSTEM MANAGEMENT

PACKETIZATION

SEGMENTATION

TRANSFER

CODING

PHYSICAL (GROUND)

TELECOMMAND

TELEMETRY

ARQ PROCEDURES

EDAC CODING PACKETS EXCHANGED

ATDRSS USER TERMINAL

ATDRSS GROUND GATEWAY

PHYSICAL LAYER SERVICE

ATDRSS SATELLITE

ATDRSS GROUND TERMINAL

NASCOM

SPACE RF LINKS

GROUND LINKS

PHYSICAL LAYER SERVICE

□ – ATDRSS RESPONSIBILITY.

SN/ATDRSS END-TO-END SERVICES

- END-TO-END SERVICE CONCEPT PROVIDES RELIABLE/STANDARDIZED/USER-FRIENDLY ACCESS TO SN
- CCSDS STANDARDS/PROTOCOLS SERVE AS REFERENCES FOR SPECIFICATION OF END-TO-END SERVICES
  - KEY CCSDS SPONSORS: NASA, ESA, NASDA
- LAYERED ATDRSS SERVICE MODEL

GRADES OF SERVICE

- 2 UNFORMATTED AND 3 FORMATTED GRADES OF SERVICE

- UNFORMATTED PHYSICAL LAYER SERVICE
  - TRANSPORT OF UNFORMATTED DATA BETWEEN POCC AND USER SPACE TERMINAL
  - UNCODED OR CONVOLUTIONALLY CODED
  - SAME AS CURRENT TDRSS

- FORMATTED SERVICE (CCSDS): END-TO-END DATA TRANSPORT VIA THREE GRADES OF PACKET-FORMATTED SERVICE
  - GRADE 1: ERROR-FREE, BLOCK-CODED, ERROR-DETECTION/CORRECTION, AUTOMATIC REPEAT REQUEST (ARQ)
    - -- E.G., SATELLITE COMMAND UPLOADING
  - GRADE 2: BLOCK-CODED, ERROR-DETECTION/CORRECTION
    - -- E.G., COMPRESSED VIDEO
  - GRADE 3: NO BLOCK CODING
  - EACH GRADE OF FORMATTED SERVICE IS TRANSMITTED VIA PHYSICAL LAYER SERVICE

TELEMETRY SERVICE DATA QUALITY REQUIREMENTS

GRADE OF SERVICE	FORMATTED SERVICE GRADE			PHYSICAL LAYER SERVICE GRADE	
	1	2	3	A (CODED)	B (UNCODED)
DATA COMPLETENESS GUARANTEED	YES	NO	NO	NO	NO
BIT ERROR RATE[1]	10^{-12}	10^{-8}	10^{-5}	10^{-5} [2]	10^{-5} [3] (TBR)
IN-SEQUENCE GUARANTEED	YES	NO	NO	N/A	N/A
DUPLICATES POSSIBLE	NO	YES	YES	NO	NO
DELIVERY OF NON-CORRECTABLE ERRORS TO USER	NO	YES	N/A	N/A	N/A
IDENTIFICATION OF NON-CORRECTABLE ERRORS TO USER	NO	YES	N/A	N/A	N/A

NOTES

1. BIT ERROR RATES INCLUDE THE OCCURRENCE OF DETECTED AND UNCORRECTED ERRORS DELIVERED TO THE USER.
2. S-BAND SERVICE ONLY.
3. KU AND Z-BAND SERVICE ONLY.

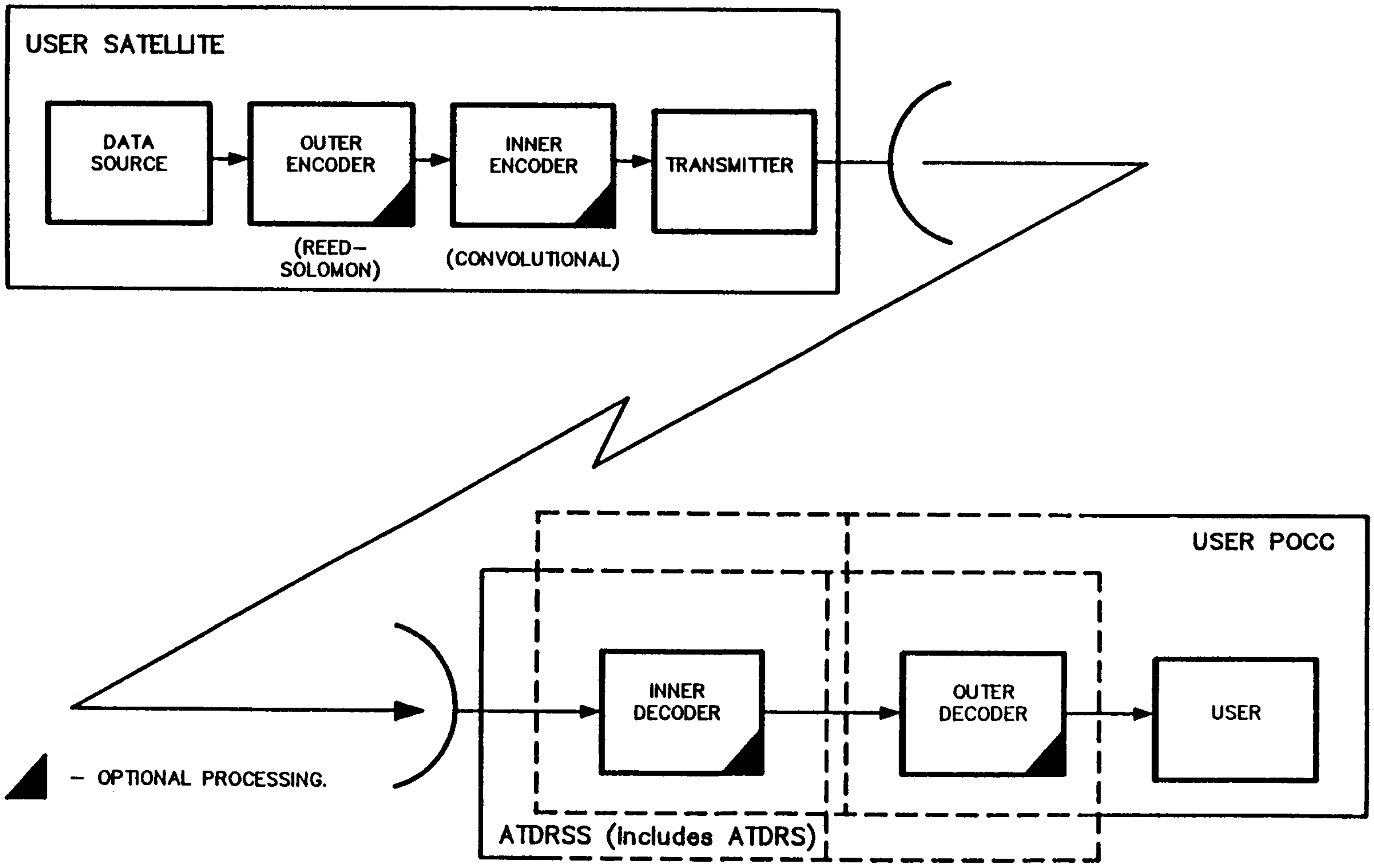
ATDRSS COMMUNICATIONS/TRACKING SERVICES*

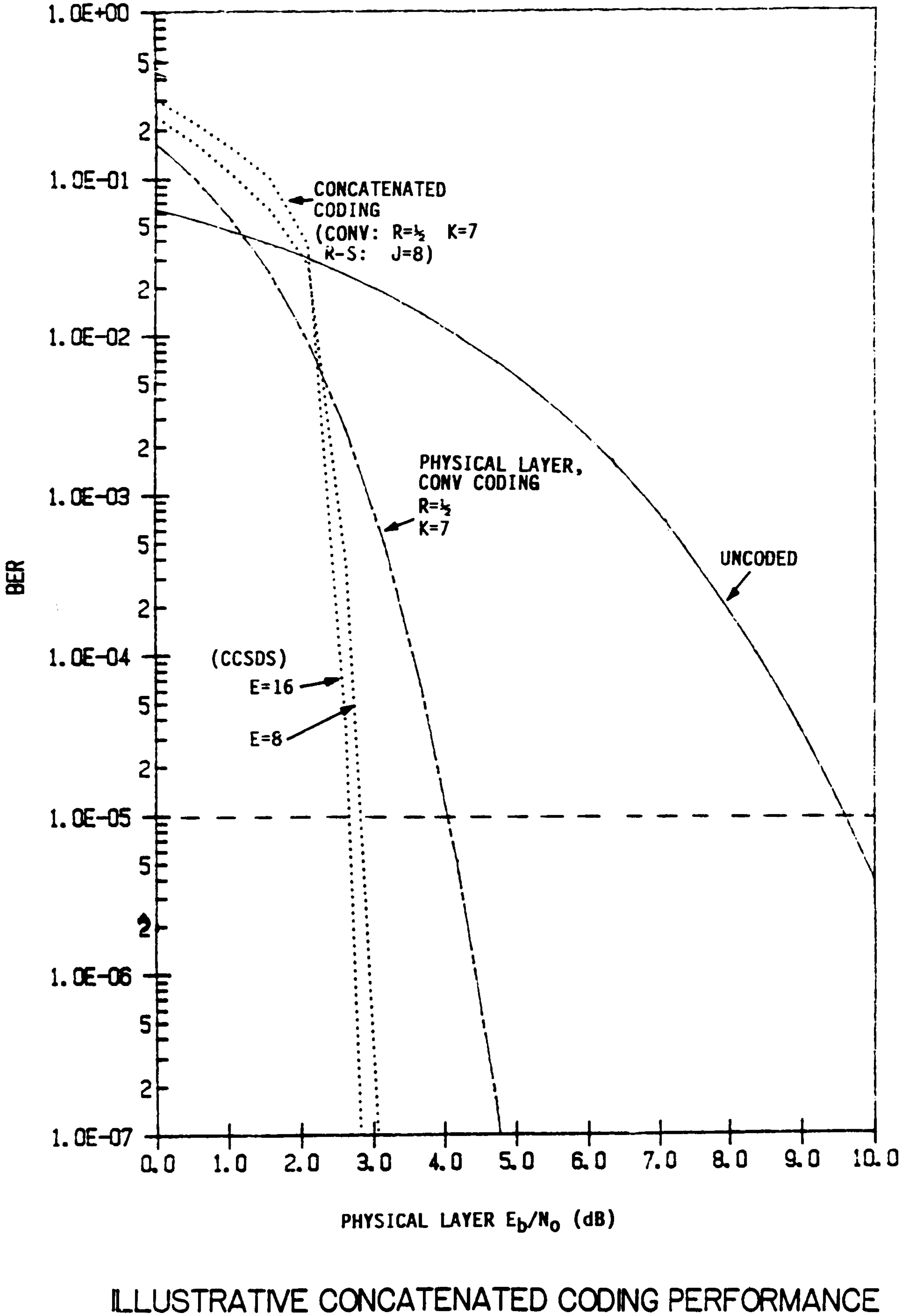
FEATURE \ SERVICE		ENHANCED SMA (EMA)	SSA	KSA	NEW HIGH DATA RATE SA (K_ASA OR WSA)
TDRSS DATA RATES RETAINED	FORWARD/ TELE-COMMAND	≤10 KBPS	• ≤300 KBPS • SHUTTLE-UNIQUE	• ≤25 MBPS • SHUTTLE-UNIQUE	—
	RETURN/ TELEMETRY	≤50 KBPS	• ≤6 MBPS • SHUTTLE-UNIQUE	• ≤300 MBPS • SHUTTLE-UNIQUE	—
NEW DATA RATES INTRO-DUCED	FORWARD/ TELE-COMMAND	—	—	• ≤50 MBPS	≤50 MBPS
	RETURN/ TELEMETRY	• ≤300 KBPS, PN • ≤3 MBPS, NO PN	—	—	≤650 MBPS
LINK QUALITY ENHANCEMENT GOALS ON PHYSICAL LAYER RELATIVE TO COMPARABLE TDRSS SERVICE		• NONE, FORWARD • 9 dB, RETURN (PROVIDES SSA LINK QUALITY)	NONE	• 3 dB, FORWARD • 4 dB, RETURN	RELATIVE TO KSA • 3 dB, FORWARD • RETURN - 9 dB, K_A SA - 8 dB, WSA
CARRIER		• 2106.4, FORWARD • 2287.5, RETURN	TUNABLE • 2030-2113, FORWARD • 2205-2295, RETURN	• 13775, FORWARD • 15003, RETURN	TBD
QUANTITY OF CHANNELS (SYSTEM)		• 4, FORWARD • 12, RETURN	4, FULL DUPLEX	8, FULL DUPLEX	4, FULL DUPLEX
SCHEDULED TRACKING SERVICES		• TWO-WAY - RANGE - DOPPLER • ONE-WAY RETURN - NONCOHERENT DOPPLER • TIME TRANSFER	→	→	→

* 2 ATDRS CONFIGURATION

TB880146/VGP4AW-1/JLT/4-14-88

CONCATENATED CODING PROVIDES PROCESSING FLEXIBILITY FOR ATDRSS





ILLUSTRATIVE CONCATENATED CODING PERFORMANCE

OUTLINE

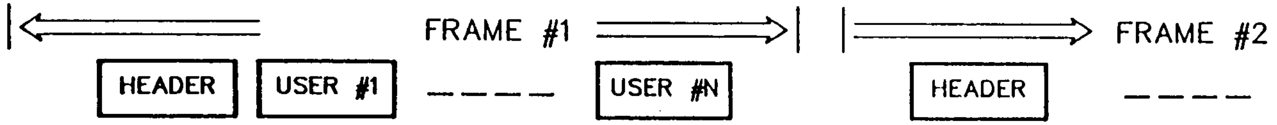
- ATDRSS DRIVERS AND OBJECTIVES
- ATDRSS ROLE WITHIN END-TO-END USER SYSTEM
- OVERVIEW OF END-TO-END SERVICES
- =======> ATDRSS SERVICE OPTIONS
- SUMMARY

REPRESENTATIVE SERVICE OPTIONS

- FULL-DUPLEX OPERATION FOR ALL EMA USERS
- NAVIGATION BEACON
- NEAR-REAL-TIME SERVICE ACCESS
- CLOSE-PROXIMITY OPERATIONS

EMA OPTION, FULL-DUPLEX SUPPORT TO ALL MA UERS

- EMA CAPABILITY PER S/C
  - 6 RETURN CHANNELS
  - 2 FORWARD CHANNELS
- UTILIZATION OF FORWARD CHANNELS
  - 1 CHANNEL PROVIDES DEDICATED SUPPORT TO ONE USER AT A TIME
    - SUPPORTS TDRSS MA USERS DURING TRANSITION
    - SUPPORTS UNIQUE USER NEEDS (e.g., COHERENT, 2-WAY TRACKING AND TIME TRANSFER)
  - TDM LINK VIA SECOND CHANNEL PROVIDES "CONTINUOUS" FORWARD CHANNELS TO EQUIPPED USERS



    - HEADER PROVIDES ORDERWIRE INFORMATION
    - 10 KBPS BURST DATA RATE
    - 100 MS AVERAGE BURST PER USER
    - ≥1-2 KBPS EFFECTIVE, CONTINUOUS DATA RATE PER USER
    - TRACKING REQUIREMENTS SATISFIED VIA COMBINATION OF NAVIGATION BEACON, ONE-WAY RETURN DOPPLER, PERIODIC UTILIZATION OF DEDICATED FORWARD CHANNEL
  - TWO-CHANNEL COMBINATION READILY SUPPORTS FULL-DUPLEX SERVICE FOR 6 USERS PER ATDRSS S/C

NAVIGATION BEACON OPTION

- BEACON TRANSMISSION VIA SINGLE S-BAND ELEMENT OF SMA ARRAY
  - 26° BEAMWIDTH
  - Ku-BAND BEACON ALSO UNDER CONSIDERATION

- AUTONOMOUS USER S/C NAVIGATION
  - PRIMARY OR BACKUP

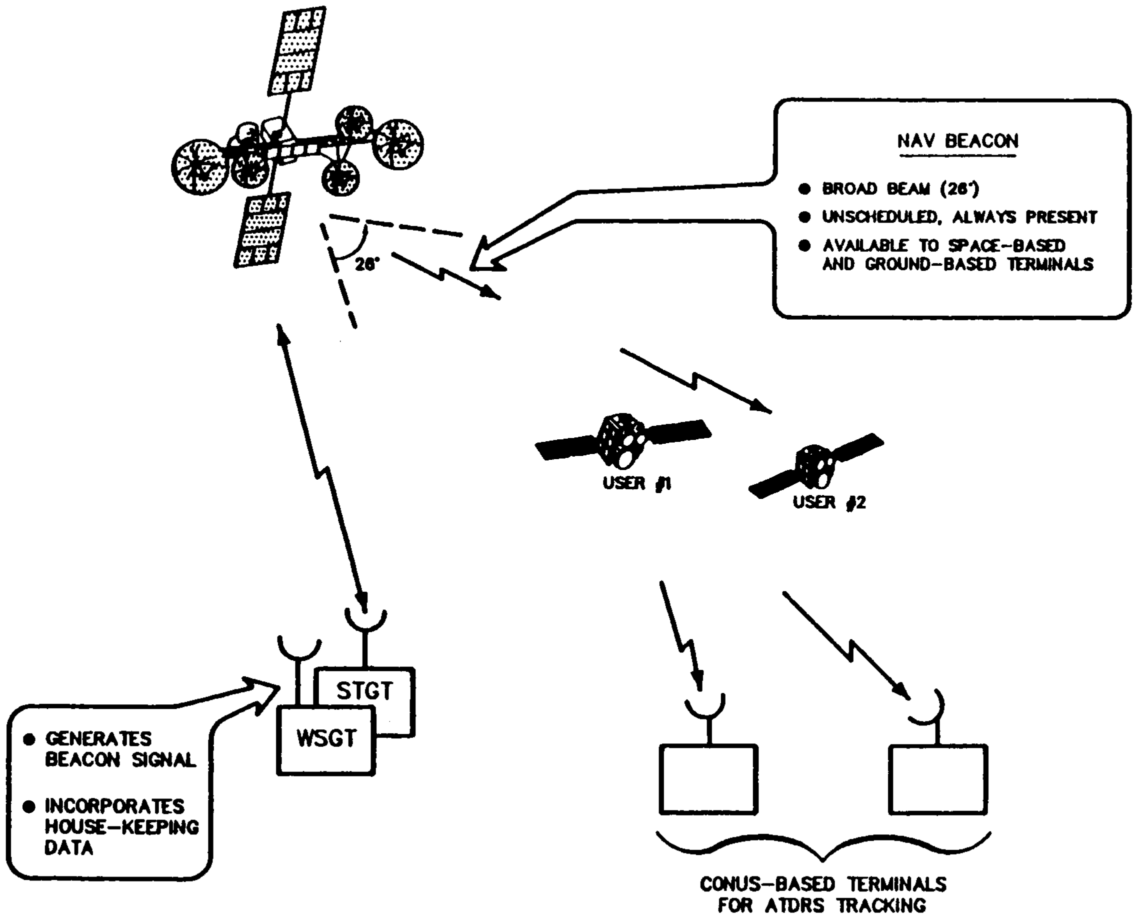
- SIMULTANEOUSLY PROVIDES CONUS-BASED ATDRS TRACKING

- UNSCHEDULED, ALWAYS PRESENT

- SATISFIES MOST USERS
  - $\leq$ 50 METERS READILY ACHIEVED
  - 10 METERS, POTENTIAL

- PROVIDES HOUSEKEEPING DATA
  - POTENTIALLY OFFERS BUILT-IN CAPABILITY FOR DEMAND ACCESS

NAVIGATION–BEACON OVERVIEW



NAV BEACON

- BROAD BEAM (26°)
- UNSCHEDULED, ALWAYS PRESENT
- AVAILABLE TO SPACE-BASED AND GROUND-BASED TERMINALS

- GENERATES BEACON SIGNAL
- INCORPORATES HOUSE-KEEPING DATA

CONUS–BASED TERMINALS FOR ATDRS TRACKING

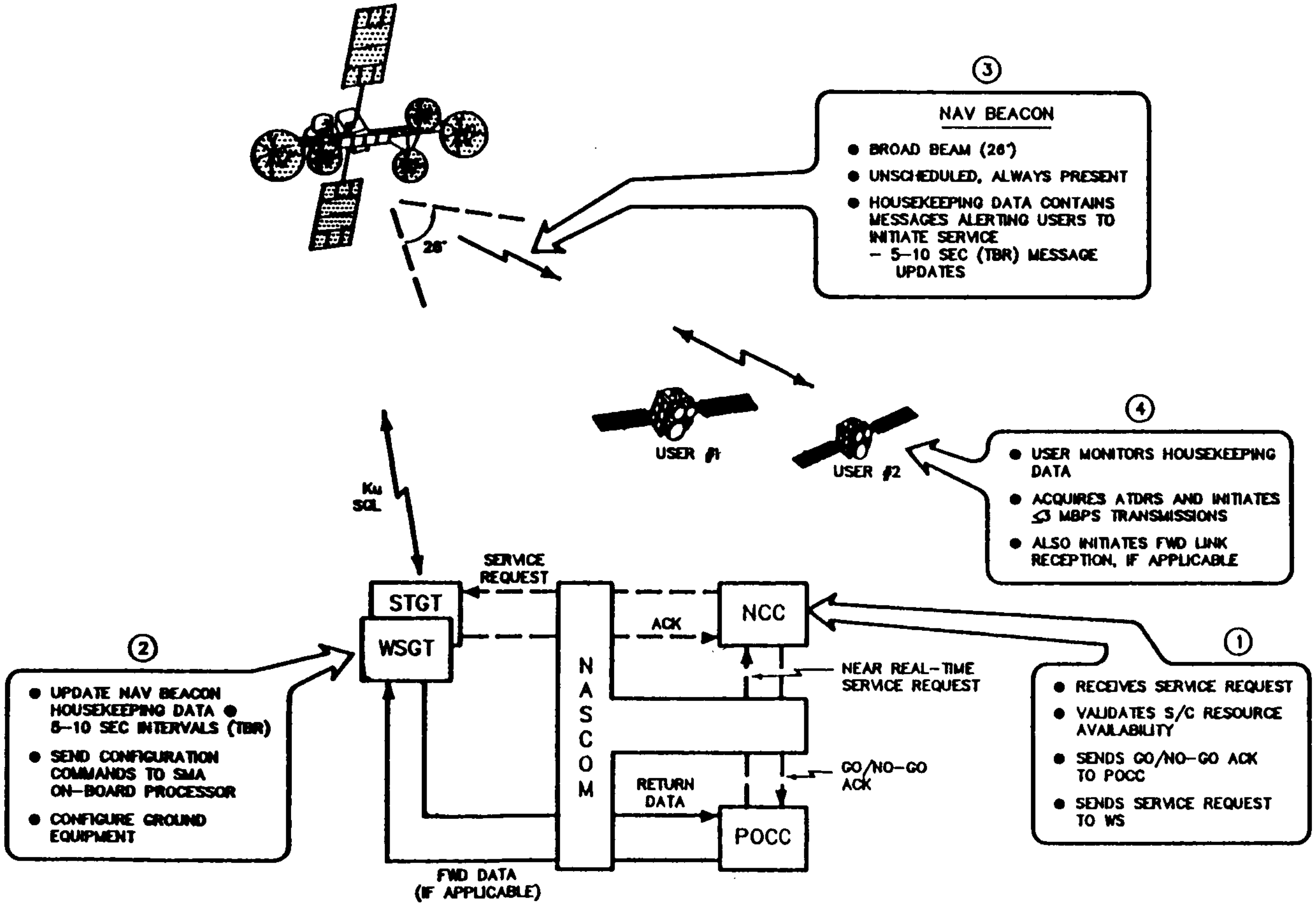
ATDRS

NEAR-REAL-TIME SERVICE ACCESS OPTION

- DEFINITION: OBTAINING "RAPID" ATDRSS SERVICE WITHOUT REQUIRING USE OF THE FORMAL, LONG-LEAD SCHEDULING PROCESS

- USAGE: ACCOMMODATE UNPLANNED/UNSCHEDULED USER SERVICE NEEDS; MAXIMIZE USER OPERATIONAL FLEXIBILITY

ILLUSTRATIVE NEAR-REAL-TIME-ACCESS SCENARIO



③

NAV BEACON

- BROAD BEAM (26°)
- UNSCHEDULED, ALWAYS PRESENT
- HOUSEKEEPING DATA CONTAINS MESSAGES ALERTING USERS TO INITIATE SERVICE
  - 5-10 SEC (TBR) MESSAGE UPDATES

④

- USER MONITORS HOUSEKEEPING DATA
- ACQUIRES ATDRS AND INITIATES ≲3 MBPS TRANSMISSIONS
- ALSO INITIATES FWD LINK RECEPTION, IF APPLICABLE

②

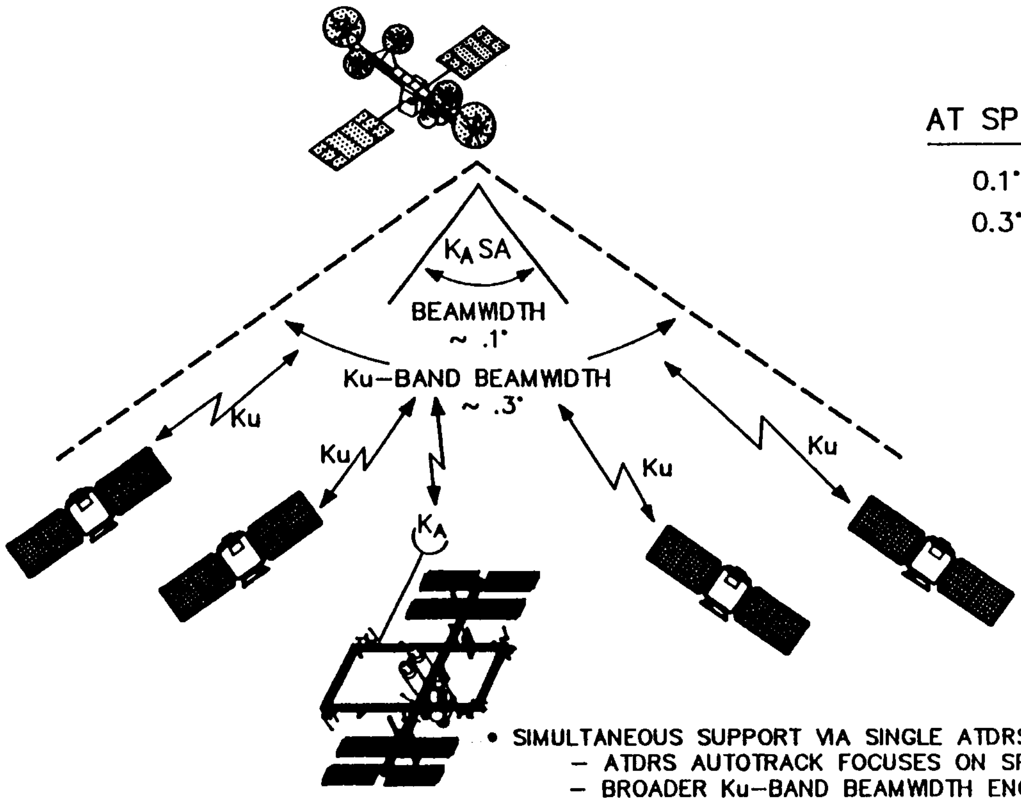
- UPDATE NAV BEACON HOUSEKEEPING DATA ● 5-10 SEC INTERVALS (TBR)
- SEND CONFIGURATION COMMANDS TO SMA ON-BOARD PROCESSOR
- CONFIGURE GROUND EQUIPMENT

①

- RECEIVES SERVICE REQUEST
- VALIDATES S/C RESOURCE AVAILABILITY
- SENDS GO/NO-GO ACK TO POCC
- SENDS SERVICE REQUEST TO WS

ATDRS

ILLUSTRATIVE CLOSE PROXIMITY OPERATIONS



AT SPACE STATION ALTITUDE:

0.1° ~ 70 km FOOTPRINT

0.3° ~ 200 km FOOTPRINT

- SIMULTANEOUS SUPPORT VIA SINGLE ATDRS Ku/K_A ANTENNA
  - ATDRS AUTOTRACK FOCUSES ON SPACE STATION
  - BROADER Ku–BAND BEAMWIDTH ENCOMPASSES OTHER CLOSE–PROXIMITY USERS
- SPACE STATION
  - FORWARD DATA RATE ≤ 50 Mbps (e.g., 2 TV CHANNELS)
  - RETURN DATA RATE, 650 Mbps (POSSIBLY HIGHER)
- AT LEAST 4 ADDITIONAL USERS
  - FDM THROUGH SINGLE KSA CHANNEL (225 MHz)
  - RETURN DATA RATE PER USER ≤ 25 Mbps (e.g., 1 TV CHANNEL)

ATDRS

OUTLINE

- ATDRSS DRIVERS AND OBJECTIVES
- ATDRSS ROLE WITHIN END-TO-END USER SYSTEM
- OVERVIEW OF END-TO-END SERVICES
- ATDRSS SERVICE OPTIONS
- =====> SUMMARY

SUMMARY

- EVOLUTION TO ATDRSS WILL OCCUR BY THE LATE 1990's
- TRANSITIONAL TRANSPARENCY FROM "TDRSS-USER" PERSPECTIVE
- ATDRSS WILL BE AN INTEGRAL PART OF END-TO-END USER DATA TRANSPORT SYSTEM
  - CCSDS STANDARDS/PROTOCOLS
  - MULTIPLE GRADES OF SERVICE
  - NEW/IMPROVED COMMUNICATIONS/TRACKING SERVICES
  - SIMPLIFIED/FRIENDLY USER INTERFACE WITH SN
- MULTIPLE SERVICE OPTIONS CAN BE ACCOMMODATED BY BASELINE ATDRSS ARCHITECTURE
- ZOE CLOSURE AND DIRECT DATA DISTRIBUTION CANNOT BE ACCOMMODATED BY BASELINE ARCHITECTURE
  - NEAR-TERM DECISION(S) REQUIRED

NASA GROUND COMMUNICATIONS

John Roeder
NASA Headquarters

ABSTRACT

As part of the Communications Requirements and Constraints, NASA's two major Ground Data Networks were briefly described.

The NASA Communication Network, called NASCOM, is the worldwide operational telecommunications system which interconnects as the tracking and telemetry acquisition sites, launch areas, mission and project control centers, data capture facilities, and network control centers in support of space flight. Currently, the NASCOM network contains over 2 1/2 million circuit miles using satellite, terrestrial, and submarine cable leased links; more than 630 circuits connect 139 domestic and foreign sites. The network is engineered and controlled at the Goddard Space Flight Center (GSFC) with major switching centers in Australia, Spain, and at the Jet Propulsion Laboratory (JPL) in California.

All kinds of communications traffic is supported, from low rate digital data and voice to narrow and wide band analog and digital at rated up to tens of megabits. NASCOM is transitioning to an all digital network with wideband links which utilizes improved technology in the competitive market place.

For the Space Station era, NASCOM plans are set for higher data rate service (up to 300 Mbps) utilizing data packet switched technology (CCSDS standards). Increased use of fiber optics is expected in a much more diverse network topology.

The second major ground network, the Program Support Communications Network (PSCN), interconnects all NASA Centers and NASA contractor

locations for intercenter non-operation communications. The primary functions are to transport voice, video, data and facsimile information for intercenter coordination, and to provide user access to space science and applications data bases. Currently, the PSCN contains almost 400 thousand circuit miles using satellite and terrestrial links, supporting many teleconference rooms and high and low speed FAX stations and a packet switched at 35 locations. This network is engineered and managed at Marshall Space Flight Center (MSFC).

Individual data rates covered are from 110 bps to 56 Kbps, 388 different types of computers and terminals are accommodated, and more than 3000 mail boxes are provided for NASA's TELEMAIL network. A computer networking subsystem allows resource sharing among the four aeronautics and space technology centers scientific mainframe computational centers.

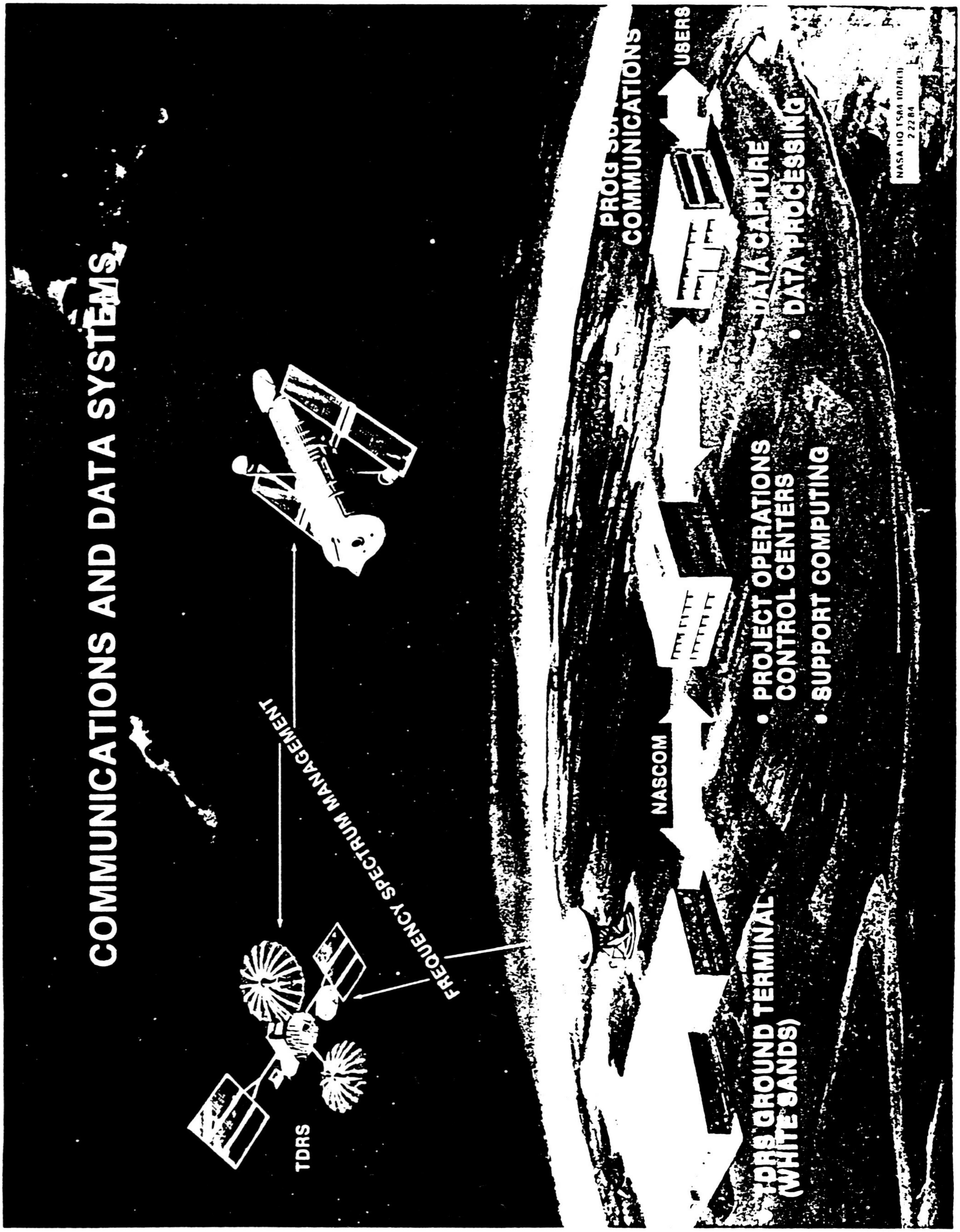
For the Space Station era, PSCN plans address the significant increase in forecast requirements for science data distribution and access to the Numerical Aerodynamics Simulator, and increased use of the Video Teleconference System.

For communications in general, a recent NASA life cycle cost analysis predicts total data volume for NASA science missions to increase as much as two orders of magnitude, by the year 2000. Obviously, costs for telecommunications will not be allowed to keep pace, so creative concepts such as data compression and information reduction are sorely needed.

NASA TELECOMMUNICATIONS

NASA OPERATIONAL COMMUNICATIONS (NASCOM)

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH



COMMUNICATIONS AND DATA SYSTEMS

NASA TELECOMMUNICATIONS

NASCOM

DEFINITION: NASCOM OPERATIONAL TELECOMMUNICATIONS INTERCONNECT NASA's FOREIGN AND DOMESTIC TRACKING AND TELEMETRY ACQUISITION SITES; LAUNCH AREAS; MISSION/PROJECT OPERATION CONTROL CENTERS; SCIENCE DATA CAPTURE FACILITIES; AND NETWORK CONTROL CENTERS.

- A PRIMARY FUNCTION IS TO TRANSPORT SPACECRAFT TELEMETERED DATA AND DATA FOR COMMAND, CONTROL, TRACKING, ORBIT DETERMINATION, AND ACQUISITION OF SPACECRAFT

- NASCOM IS EXEMPT FROM THE FIRMR; HOWEVER, NASCOM FOLLOWS ALL APPLICABLE PROCUREMENT REGULATIONS IN ACQUISITION PROCESS

- NASCOM IS A MAJOR NATIONAL COMMUNICATIONS SYSTEM (NCS) ASSET

- MANAGED AND CONTROLLED BY THE GODDARD SPACE FLIGHT CENTER

NASA COMMUNICATIONS

NASCOM - CURRENT DESCRIPTION

- o OVER 2 1/2 MILLION CIRCUIT MILES - DOMESTIC AND FOREIGN; SATELLITE, TERRESTRIAL, AND SUBMARINE CABLE
- o 630 CIRCUITS TO 139 STATIONS
  - VOICE, DATA, VIDEO
  - ANALOG, DIGITAL
  - NARROW BAND, WIDE BAND
- o 35 COLLOCATED DOMSAT STATIONS AT 11 NASA SITES
- o BROADCAST MODE VIDEO SYSTEM TO ALL NASA CENTERS
- o CONTROL CENTER - GSFC
- o MAJOR SWITCHES AT AUSTRALIA, SPAIN, AND JPL
- o NEARLY ALL LEASED CIRCUITS

NASA TELECOMMUNICATIONS

FUTURE COST GROWTH

- o NASCOM WILL INCREASE SLIGHTLY BECAUSE OF INCREASED BANDWIDTH FOR SPACE STATION (LESS THAN 30%).

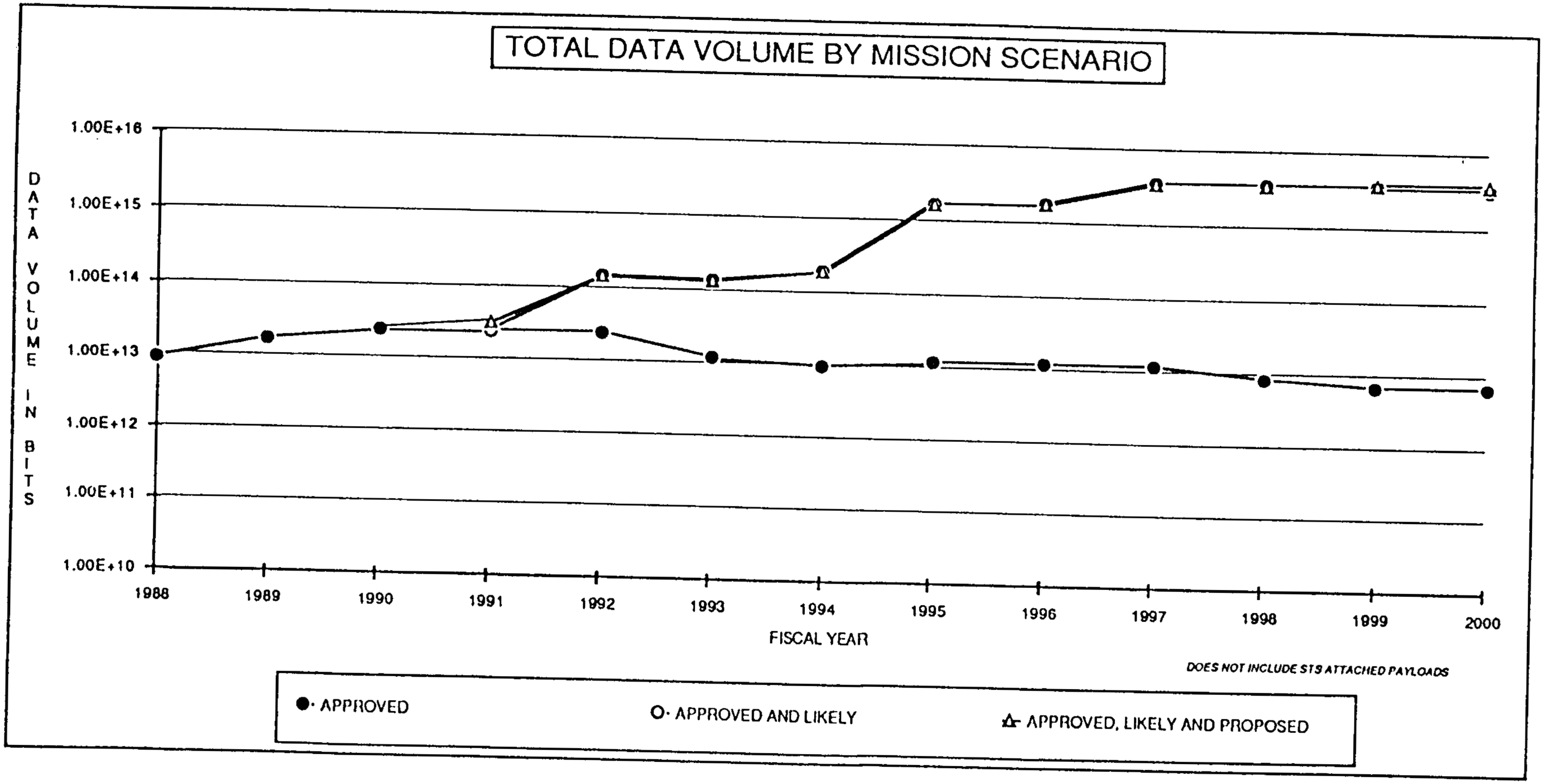
- o COSTS IN TELECOMMUNICATIONS TEND TO REMAIN STABLE BECAUSE OF COMPETITIVENESS IN MARKET PLACE. THE COST PER BIT CONTINUES TO DECREASE.

- o DATA VOLUME EXPECTED TO INCREASE BY TWO ORDERS OF MAGNITUDE; HOWEVER, BUDGETED COSTS MUST REMAIN BELOW A 30% GROWTH.

- o DATA COMPRESSION, INFORMATION REDUCTION, OR CREATIVE CONCEPTS WILL BE REQUIRED.

Taken From: Life Cycle Cost Analysis for NASA Science Data Handling 1988-2000. Dec. 23, 1987

A report to the HUD-Independent Agencies Subcommittee of the Committee on Appropriations of the United States Senate.



TOTAL DATA VOLUME BY MISSION SCENARIO

NASA COMMUNICATIONS

NASCOM NEAR-TERM PLANNING FOR SPACE STATION ERA

- o INCREASED WIDEBAND DATA SYSTEM USING DATA PACKET SWITCHED TECHNOLOGY (CCSDS)
- o AVERAGE DATA RATES ON BASELINE SYSTEM APPROACHING 90 MBS FROM WHITE SANDS TO JSC, GSFC, MSFC, WEST COAST
- o MAXIMUM DATA RATE OF 300 MBS WHEN REQUIRED
- o NETWORK TOPOLOGY BEING STUDIED TO USE ALL FIBER OPTICS TECHNOLOGY
- o COST EFFECTIVENESS ARCHITECTURE UNDER REVIEW

NASA TELECOMMUNICATIONS

NASCOM - TRANSITIONING TO:

- A COMPLEX COMMON USER NETWORK COMPRISED OF WIDEBAND DIGITAL LINKS
- THESE LINKS WILL:
  - PROVIDE ACCOMMODATION FOR TDRSS BASELINE (4-10 MBS)
    TDRSS WIDEBAND (50 MBS)
  - UTILIZE TDMA TECHNIQUES FOR DEMAND ASSIGNED VOICE, HIGH SPEED DATA AND VIDEO
  - UTILIZE FIBER OPTICS FOR WIDEBAND INTERFACES AND LOCAL DISTRIBUTION
- CAN EXPECT LONG TERM STABILIZATION OF OPERATIONS
  - IMPROVED TECHNOLOGY
  - COMPETITIVE MARKET PLACE

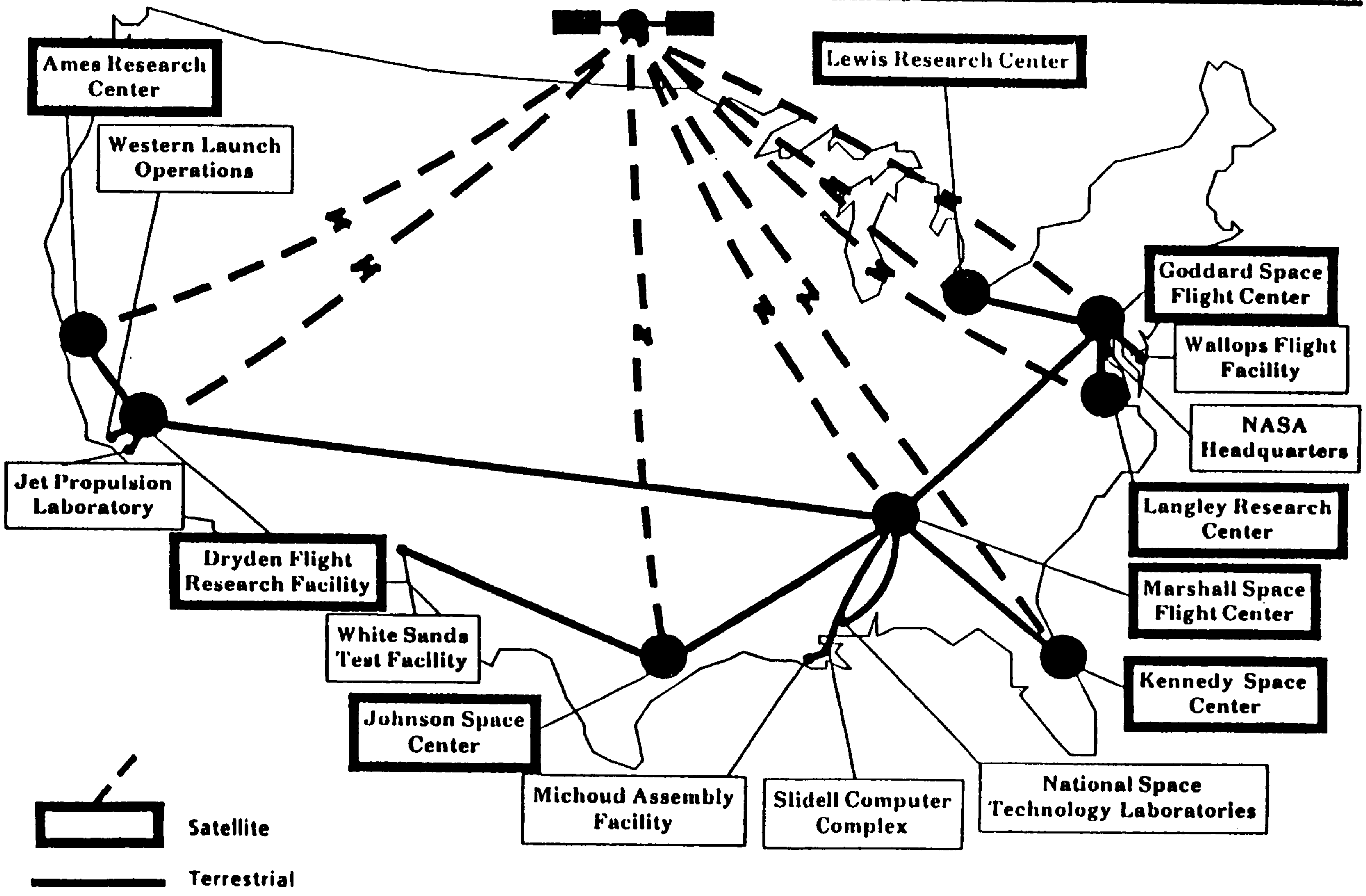
NASA TELECOMMUNICATIONS

PROGRAM SUPPORT COMMUNICATIONS (PSC)

DEFINITION: PROGRAM SUPPORT COMMUNICATIONS INTERCONNECT ALL NASA CENTERS AND NASA CONTRACTOR LOCATIONS FOR INTERCENTER NON-OPERATIONAL COMMUNICATIONS.

- A PRIMARY FUNCTION IS TO TRANSPORT VOICE, VIDEO, DATA, AND FACSIMILE INFORMATION FOR INTERCENTER COORDINATION FUNCTION AS WELL AS USER ACCESS TO NASA MAINTAINED SPACE SCIENCE AND APPLICATIONS DATA BASES
- PSC FOLLOWS ALL APPLICABLE GSA FIRMR'S AND FPR's
- PSC IS A NATIONAL COMMUNICATIONS SYSTEM (NCS) ASSET UNDER GSA
- PSC IS MANAGED BY THE MARSHALL SPACE FLIGHT CENTER

Program Support Communications - Backbone Network


Ames Research Center
Western Launch Operations
Lewis Research Center
Goddard Space Flight Center
Wallops Flight Facility
NASA Headquarters
Jet Propulsion Laboratory
Langley Research Center
Dryden Flight Research Facility
Marshall Space Flight Center
White Sands Test Facility
Kennedy Space Center
Johnson Space Center
Michoud Assembly Facility
Slidell Computer Complex
National Space Technology Laboratories
Satellite
Terrestrial

NASA-PSC

NASA TELECOMMUNICATIONS

PSC - CURRENT DESCRIPTION

- 377,000 CIRCUIT MILES - BOTH SATELLITE AND TERRESTRIAL
- 84 VOICE TELECONFERENCE ROOMS
  - 4632 VOICE TELECONFERENCES IN FY 83
- 152 HI SPEED FAX STATIONS
- 156 LO SPEED FAX STATIONS
- PACKET SWITCHED NETWORK AT 35 LOCATIONS
  - DATA FOR SHUTTLE, LEGAL, MEDIA, IG
- UTILIZES
  - FACILITIES OF 14 COMMON CARRIERS IN 17 STATES
  - FACILITIES OF 6 SPECIALIZED COMMON CARRIERS IN 6 STATES
  - 388 DIFFERENT TYPES OF COMPUTERS AND TERMINALS
- DATA RATES FROM 110 B/S TO 56 KB/S
- 3133 MAILBOXES ON TELEMAIL NASANET
- NETWORK CONTROL AT MSFC

NASA TELECOMMUNICATIONS

PSC PROCUREMENT

- CONSOLIDATED, IN A SINGLE PROCUREMENT, SEVERAL EXISTING CONTRACTS AND PLANNED PROCUREMENTS
- THE ELEMENTS WERE COMBINED TO ACHIEVE BETTER SYSTEMS ECONOMIES, EFFICIENCIES, COORDINATION, AND CONTROL THROUGH A SINGLE MANAGEMENT FOCUS.
- SINGLE CONTRACT FOR ALL ELEMENTS

NASA TELECOMMUNICATIONS

PSC CONTRACT ELEMENTS

- PROGRAM SUPPORT COMMUNICATIONS NETWORK (PSCN)- INTEGRATED COMMON USER NETWORK
- COMPUTER NETWORKING SUBSYSTEM - PROVIDES CAPABILITY OF THE 4 AERONAUTICS AND SPACE TECHNOLOGY CENTERS SCIENTIFIC MAINFRAME COMPUTATIONAL COMPUTERS FOR RESOURCE SHARING
- MARSHALL SPACE FLIGHT CENTER (MSFC) LOCAL TELEPHONE SYSTEMS (HUNTSVILLE, SLIDELL COMPUTER COMPLEX, MICHOUD ASSEMBLY FACILITY) - DIGITAL VOICE AND DATA SYSTEMS TO REPLACE AND UPGRADE THE PRESENT TELEPHONE COMPANY SYSTEMS.
- MSFC TELECOMMUNICATIONS MISSION SERVICES - ENGINEERING AND OPERATIONAL SERVICES TO SUPPORT MSFC INTRA CENTER COMMUNICATIONS
- FTS - INTERCITY VOICE SERVICE BETWEEN NASA CENTERS.

NASA TELECOMMUNICATIONS

PSC CONTRACT

- AWARDED TO BOEING COMPUTER SUPPORT SERVICES (BCSS) - MARCH 31, 1985
- BCSS SUBCONTRACTORS - BAMSI, RCA SERVICE COMPANY
- 5 YEAR CONTRACT WITH OPTION FOR AN ADDITIONAL 5 YEARS

NASA TELECOMMUNICATIONS

PSCN CHARACTERISTICS

- NO SINGLE POINT OF FAILURE
- ALTERNATE ROUTING
- DES ENCRYPTION GATEWAY TO GATEWAY
- USE TDRSS C-BAND - IF APPROVED
- STRONG NCC AT MSFC FOR END TO END TEST AND DIAGNOSTICS

NASA TELECOMMUNICATIONS

PSCN NEAR-TERM PLANNING FOR SPACE STATION ERA

- o SIGNIFICANT INCREASE IN FORECASTED REQUIREMENTS
- o MAJOR REQUIREMENTS AND COST GROWTH
    - o SCIENCE DATA DISTRIBUTION
    - o NUMERICAL AERODYNAMIC SIMULATOR (NAS)
        - - ACCESS TO NAS FROM NASA AND NASA SPONSORED LOCATIONS
- o INCREASED USE OF VIDEO TELECONFERENCING SYSTEM

"DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL"

Edward C. Posner
Jet Propulsion Laboratory

ABSTRACT

Deep space telemetry is and will remain signal-to-noise limited and vulnerable to interference. We do all we can to increase received signal power and decrease noise. This includes going to Ka-band (32 GHz down) in the mid-1990's to increase directivity. This is in spite of the increased difficulty of maintaining surface accuracy, pointing the spacecraft and ground antennas, and accommodating to weather uncertainty. The effects of a wet atmosphere can increase the noise temperature by a factor of 5 or more, even at X-band (8.5GHz down), but the order of magnitude increase in average data rate obtainable at Ka-band relative to X-band makes the increased uncertainty a good trade. The 32GHz frequency is likely to be the highest frequency used operationally from deep space in the next 15 to 20 years. Lowbit error probabilities required by data compression are available both theoretically and practically with coding, at an infinitesimal power penalty (.05-.2dB) rather than the 10-15dB more power required to reduce error probabilities without coding. Advances are coming rapidly in coding, as with the new constraint-length 15 rate 1/4 convolutional code concatenated with the already existing Reed-Solomon code to be demonstrated on Galileo. These advances will get NASA ready for the day when high-compression-ratio telemetry will require 10^{-6} to 10^{-9} bit error probability. In addition, high density spacecraft data storage will allow selective retransmissions, even from the edge of the Solar System, to overcome weather effects. In general, deep space communication has been able to operate, and will continue to operate, closer to theoretical limits than any other form of communication. These include limits in antenna area and directivity, system noise temperature, coding efficiency, and

everything else. The deep space communication links of the mid-90's and beyond will be compatible with new instruments and compression algorithms and represent a sensible investment in an overall end-to-end information system design.

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

BY EDWARD C. POSNER

TELECOMMUNICATIONS AND DATA ACQUISITION OFFICE
JET PROPULSION LABORATORY

PRESENTED AT NASA SCIENTIFIC DATA COMPRESSION WORKSHOP
SNOWBIRD, UTAH

MAY 3, 1988

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

BY EDWARD C. POSNER, JPL

MAY 3, 1988

OUTLINE

- UNIQUENESS OF DEEP SPACE COMMUNICATION
- NOISE AS A LIMITING FACTOR
- FREQUENCY SELECTION
- WEATHER EFFECTS
- CHANNEL CODING AND DATA COMPRESSION
- VISIBLE FUTURE TELEMETRY TRENDS
- SUMMARY

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

UNIQUENESS OF DEEP SPACE COMMUNICATIONS

- WEAK SIGNALS
  - SIGNALS FROM EDGE OF SOLAR SYSTEM TO EARTH SUFFER MORE THAN 10 BILLION TIMES AS MUCH PATH LOSS AS THOSE FROM SYNCHRONOUS ORBIT
  - IMPLIES CONSERVING SIGNAL POWER MORE IMPORTANT THAN OTHER CONSTRAINTS, E.G., BANDWITH
  - MAKES DEEP SPACE SIGNALS MORE VULNERABLE TO INTERFERENCE
- LONG PROPAGATION TIMES
  - ONE-WAY TIME FROM EDGE OF SOLAR SYSTEM 4 HOURS INSTEAD OF 0.1 SEC FROM SYNCHRONOUS ORBIT
  - IMPLIES ONE-WAY BULK DATA TRANSMISSION APPROPRIATE WITH LITTLE USE OF FEEDBACK
  - MAKES IT HARD TO ADJUST SPACECRAFT BASED ON CONDITIONS AT EARTH

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

NOISE AS A LIMITING FACTOR

- CHANNEL CAPACITY DETERMINED BY RECEIVED SIGNAL POWER, NOISE POWER, AND (LESS IMPORTANT FOR DEEP SPACE) BANDWIDTH
- NOISE POWER IS RECEIVED FROM
  - COSMIC BACKGROUND
  - PLANET IN BEAM, ESP. VENUS
  - EARTH's ATMOSPHERE (WET IS WORSE)
  - EARTH ITSELF (SIDELOBES, BACKLOBES)
  - RECEIVER FRONT END
- NOISE POWER SPECTRUM TENDS TO BE FLAT (WHITE) ACROSS BAND, OF HEIGHT OR POWER SPECTRAL DENSITY $N_0 = kT$
  - k IS BOLTZMAN'S CONSTANT, T IS SYSTEM TEMPERATURE IN KELVINS
- CHANNEL CAPACITY C IN BITS/SEC IS THEN
  - $C = P_R / N_0$ 1n2 (P_R IS POWER RECEIVED)

CONTROLLING NOISE

- 3 K COSMIC BACKGROUND AND PLANET IN BEAM ARE NOT CONTROLLABLE NOISE SOURCES
- EARTH's WET ATMOSPHERE DEVASTATING BECAUSE SIGNAL ABSORBTION EQUIVALENT TO NOISE RADIATION
  - EXAMPLE: 10% ABSORPTION SEEMS MINOR (0.46 dB LOSS), BUT IMPLIES 10% NOISE RADIATION AT SAY 290 K: 29 K TEMPERATURE INCREASE, AROUND A 3 dB LOSS
  - CONCLUSION: WEATHER IS RISK ABOVE S-BAND (2 GHz)
- EARTH RADIATION INTO ANTENNA CAN BE REDUCED BY QUADRUPOD DESIGN, BECOMES LESS THAN 3 K BACKGROUND
- RECEIVER FRONT-END OPERATES COLD
  - DEPENDING ON QUANTUM-MECHANICAL PRINCIPLE, CAN OPERATE WARMER THAN NOISE TEMPERATURE CONTRIBUTION, E.G., HEMT's

FREQUENCY SELECTION

- HIGHER MICROWAVE FREQUENCIES WITH DIRECTIONAL ANTENNAS RESULT IN MORE RECEIVED SIGNAL POWER, PROPORTIONAL TO SQUARE OF FREQUENCY
  - ANTENNA POINTING HARDER; NEEDED ANGULAR ACCURACY LINEAR IN FREQUENCY FOR FIXED ANTENNA DIAMETERS ON SPACECRAFT AND GROUND
  - SURFACE ACCURACY REQUIRED LINEAR IN FREQUENCY
  - WEATHER EFFECTS ON SYSTEM TEMPERATURE GENERALLY MORE PRONOUNCED AT HIGHER MICROWAVE FREQUENCIES
  - WIND ALSO IMPORTANT FOR GROUND ANTENNA POINTING AT LARGER DIAMETERS AND HIGHER FREQUENCIES
  - PLANET IN RECEIVE BEAM MORE IMPORTANT NOISE SOURCE AS BEAM NARROWS WITH HIGHER FREQUENCIES

- AT VENUS, EXCESS NOISE AT 2 GHz, 70-METER, VENUS CLOSEST APPROACH, IS ABOUT 11 K, BUT AT 32 GHz VENUS FILLS BEAM, 600 K NOISE TEMPERATURE INCREASE

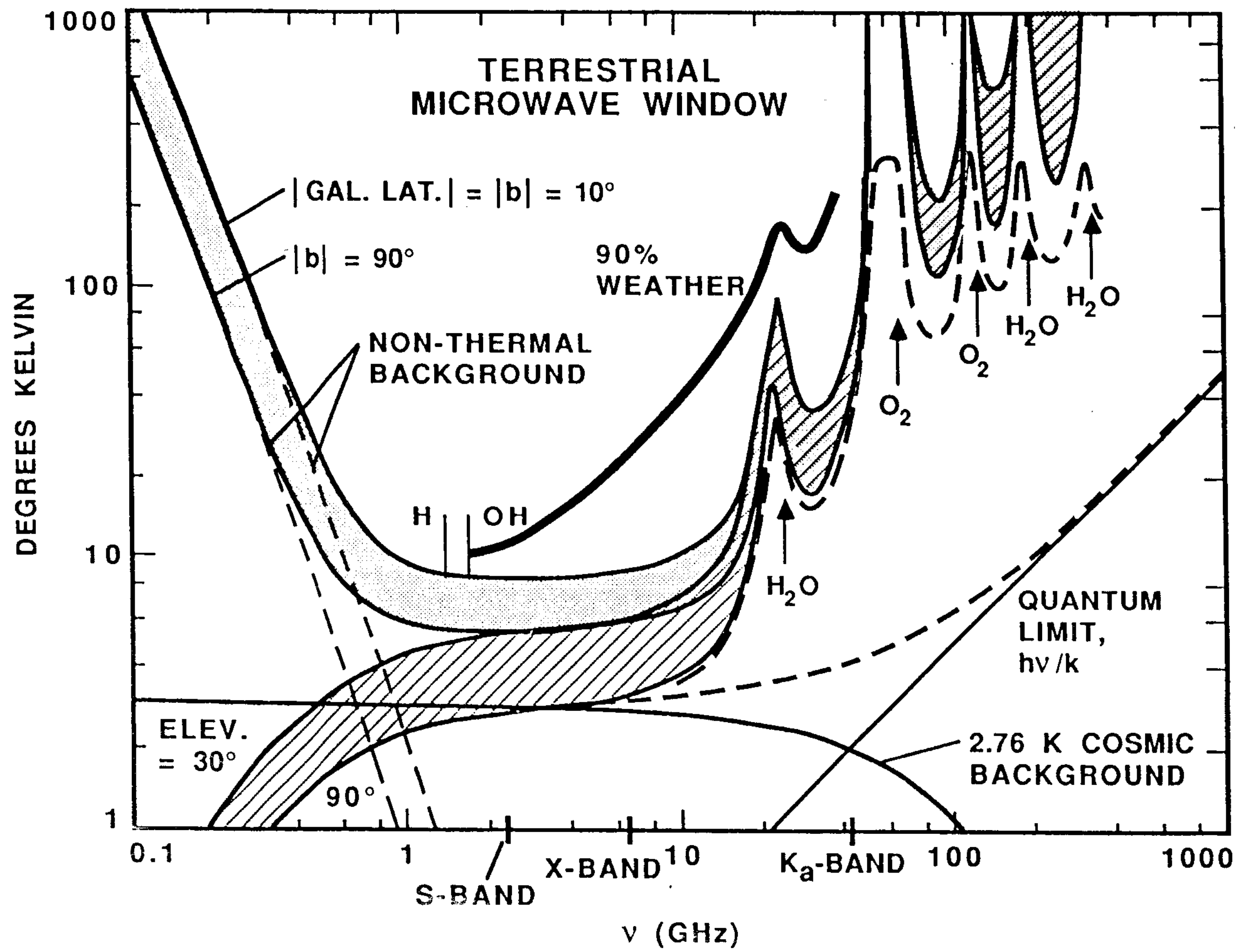
- THUS, ALL OTHER EFFECTS TEND TO REDUCE DATA RATE AS FREQUENCY INCREASES

- NEVERTHELESS, CAREFUL ENGINEERING AND WEATHER STRATEGY MAKE HIGHER FREQUENCIES FOR DEEP SPACE TELEMETRY GOOD INVESTMENT
  - K_a -BAND (32 GHz) LIKELY TO BE HIGHEST FREQUENCY FOR ROUTINE ON-EARTH RECEPTION FOR THREE DECADES

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL



TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

WEATHER EFFECTS

- UNCERTAINTY IN WEATHER AND LONG PROPAGATION DELAYS CAUSE SOME PROBABILITY OF LOSS OF DEEP SPACE DATA AT X-BAND (8.5 GHz) AND HIGHER FREQUENCIES
    - LINK OPERATES WITH STEEP PERFORMANCE CURVE
    - SIGNAL PAD REDUCES AVERAGE DATA RATE
    - WEATHER MODELS NEEDED TO ASSESS RISK
    - VOYAGER AT URANUS AND NEPTUNE WORKED/WILL WORK AT 90% WEATHER CONFIDENCE

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

WEATHER EFFECTS (Cont'd)

- WEATHER EFFECTS COULD BE PARTIALLY MITIGATED IN FUTURE BY
  - "BROADCAST" CODING SCHEME WHICH STILL RETURNS SOME DATA EVEN IN ADVERSE WEATHER
  - BETTER WEATHER MODELS
  - MORE ANTENNA DIVERSITY ON GROUND
  - MASSIVE SPACECRAFT DATA STORAGE WITH SELECTIVE RETRANSMISSION
  - RETARGETTING ORBITERS TO FILL IN MISSED AREAS
- WEATHER EFFECTS COULD BE COMPLETELY ELIMINATED BY AN EARTH-ORBITING DEEP SPACE RELAY STATION
- PLANNED MODE OF COPING WITH WEATHER FOR NEXT 20 YEARS INVOLVES
  - BETTER WEATHER MODELS
  - SPACECRAFT DATA STORAGE DEVELOPMENT

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

CHANNEL CODING AND DATA COMPRESSION

- SHANNON TELLS US THAT A LARGE NUMBER OF BITS SHOULD BE ENCODED INTO A LONG BLOCK OR STRING AND THE RECEIVED WAVEFORM DETECTED AS A UNIT
  - EXTRA POWER THEORETICALLY NEEDED TO DROP BIT ERROR PROBABILITY FROM 10^{-3} TO "ZERO" IS ONLY 0.05 dB
- CODING GAIN RELATIVE TO NO CODING INCREASES AS DESIRED BIT ERROR PROBABILITY DROPS
  - AT 10^{-6} BIT ERROR PROBABILITY, THEORETICAL CODING GAIN ACHIEVABLE IS AROUND 12 dB
- LOW BIT ERROR PROBABILITY FROM DEEP SPACE BECOMING MORE DESIRABLE DUE TO
  - MORE EXTENSIVE USE OF DATA COMPRESSION
  - TREND TOWARD AUTOMATED GROUND OPERATIONS
  - MORE SPACECRAFT INTELLIGENCE WITH AFTER-THE-FACT REPORTING

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

CHANNEL CODING AND DATA COMPRESSION (Cont'd)

- CODING COULD INTERACT MORE CLOSELY WITH INSTRUMENTS IN OTHER WAYS THAN JUST THROUGH LOW BIT ERROR PROBABILITY, BY
  - USING SOURCE STATISTICS TO IMPROVE DECODER PERFORMANCE AS IN VOYAGER NEPTUNE VITERBI DECODER BACKUP
  - COMBINED COMPRESSION AND CODING
    - SHANNON TELL US THAT THEORETICAL POWER EFFICIENCY DOES NOT DROP IF COMBINE THE TWO
    - COMPLEXITY CAN DROP GREATLY

C-3

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

VISIBLE FUTURE TELEMETRY TRENDS

- CODING CLOSER TO SHANNON LIMIT
  - GALILEO CONSTRAINT-LENGTH 15 RATE 1/4 CONVOLUTIONAL CODING DEMO LEADING TO NEW NASA STANDARD CONSTRAINT-LENGTH 15 RATE 1/6 CONVOLUTIONAL CODE
- HIGHER FREQUENCIES: 32 GHz DOWN (K_a-BAND) FOR CASSINI AND MARS ROVER SAMPLE RETURN
- MORE DATA COMPRESSION REQUIRING LOWER BIT ERROR PROBABILITIES
- MUCH MORE MASSIVE SPACECRAFT DATA STORAGE BY LATE 1990's PARTIALLY ALLEVIATING WEATHER EFFECTS PROBLEM
- DEMAND FOR MUCH HIGHER DATA RATES TO ACCOMMODATE INSTRUMENTS SUCH AS IMAGING SPECTROMETERS
  - STILL NEED VERY LOW BIT ERROR PROBABILITIES
  - DECODERS WILL HAVE TO OPERATE FAST
- BEGINNING OF DEEP SPACE TELEMETRY OPERATING NEAR BANDWIDTH LIMITS
  - CONSERVING SIGNAL POWER WILL STILL BE IMPORTANT
  - SUGGESTS COMBINED MODULATION AND CODING

TELECOMMUNICATIONS AND DATA ACQUISITION

DEEP SPACE COMMUNICATIONS, WEATHER EFFECTS, AND ERROR CONTROL

JPL

SUMMARY

- DEEP SPACE TELEMETRY IS AND WILL REMAIN SIGNAL-TO-NOISE LIMITED
- WE DO ALL WE CAN TO INCREASE RECEIVED SIGNAL POWER AND DECREASE NOISE
- WEATHER GREATLY INCREASES NOISE UNPREDICTABLY AT DEEP SPACE FREQUENCIES
- LOW BIT ERROR PROBABILITIES AT HIGH POWER EFFICIENCY OBTAINABLE BY CODING
- FUTURE DEEP SPACE TELEMETRY WILL HAVE EXQUISITELY CODED HIGH BIT RATE HIGHLY COMPRESSED DATA WITH SELECTIVE RETRANSMISSION TO OVERCOME WEATHER EFFECTS
- DEEP SPACE COMMUNICATION OPERATES CLOSER TO THEORETICAL LIMITS THAN ANY OTHER FORM OF COMMUNICATION

COMPRESSION METHODS

OVERVIEW

TOM STOCKHAM

UNIVERSITY OF UTAH

SESSION COORDINATOR

SUBJECTIVE vs. OBJECTIVE ISSUES in DATA COMPRESSION

Tom Stockham
University of Utah

Abstract not available.

LOSSLESS CODING METHODS

Robert Rice
Jet Propulsion Laboratory

Abstract not available. Presentation covered material from references cited below which can be requested from the author.

REFERENCES

1) Rice, R.F., A.P. Schlutsmeyer, "Data Compression for NOAA Weather Satellite Systems," SPIE Seminar Proceedings, Vol. 249, July 1980.

2) Rice, R.F., "Block Adaptive Rate Controlled Image Data Compression," Proceedings 1979 National Telecom. Conf., Washington, D.C., Nov. 1979.

3) Rice, R.F., " End-to-End Imaging Information Rate Advantages of Various Alternative communication Systems," JPL Publication 82-61, JPL, Pasadena, CA., Sept. 1, 1982.

4) Rice, R.F., J. Lee, "Noiseless Coding for the Gamma Ray Spectrometer," JPL Publication 85-53, JPL, Pasadena, CA, June 1985.

5) Rice, R.F., J. Lee, "Noiseless Coding for the Magnetometer," JPL Publication 87-19, JPL, Pasadena, CA, June 15, 1987.

6) Rice, R.F., J. Lee, "Some Practical Universal Noiseless Coding Techniques Part II," JPL Publication 83-17, JPL, Pasadena, CA, March 1, 1983.

7) Rice, R.F., "Some Practical Universal Noiseless Coding Techniques," JPL Publication 79-22, JPL, Pasadena, CA, March 15, 1979.

8) Rice, R.F. "The Development of Efficient Coding for an Electronic Mail System," JPL Publication 83-64, JPL, Pasadena, CA, July 15, 1983.

LOSSY CODING METHODS

Anil Jain
Optivision

Abstract not available.

VECTOR QUANTIZATION

Robert M. Gray
Information Systems Laboratory

ABSTRACT

During the past ten years Vector Quantization (VQ) has developed from a theoretical possibility promised by Shannon's source coding theorems into a powerful and competitive technique for speech and image coding and compression at medium to low bit rates. In this survey, the basic ideas behind the design of vector quantizers are sketched and some comments made on the state-of-the-art and current research efforts.

INTRODUCTION

VQ can be thought of as the vector extension of Pulse Coded Modulation (PCM) wherein real vectors instead of real scalars are converted into digital representations which in turn can be used to produce a reproduction of the original signal. The goal, of course, is to produce a digital representation of the signal which can be communicated on a digital communication channel or stored in a digital medium. Representing analog data digitally introduces distortion, and hence a major design goal is to minimize the distortion given constraints on communication or storage capacity and complexity. The vectors to be digitized may be a collection of consecutive samples from a continuous waveform, rectangular subblocks of an image intensity or density, three dimensional vectors consisting of, say, two-by-two squares of pixels three frames deep (or twelve pixels in the vector), or they may be feature or parameter vectors extracted from the data which represent its important attributes, such as Fourier transformed vectors or the Linear Predictive Coded (LPC) representation of a speech signals.

Although Shannon's source coding theorems imply that performance improvement can always be obtained by coding vectors instead of scalars [1,2,3,55,62], the most popular systems for analog-to-digital conversion and data compression perform the quantization operation only on scalars, although they often effectively operate on vectors by imbedding the quantizer in a feedback loop (as in predictive quantization) or by first performing a linear transform on the data (as in transform coding). While such systems have the advantage of simplicity, they are necessarily suboptimal in the Shannon sense. Furthermore, the definition of "simplicity" has been enormously extended with modern circuit design and implementation techniques: DSP chips and VLSI have rendered feasible algorithms that were considered absurdly complicated only ten years ago.

In addition to the complexity issue, another impediment to the use of VQ systems has been the lack of design algorithms. Unlike the dual problem of channel coding or error control coding, quantization is fundamentally nonlinear and the algebraic approaches successful in error correction are of little help in the digitization problem. Since the middle of the last decade, a variety of design techniques and tricks have been developed for VQ and real time hardware has been designed to perform sophisticated variations of these systems. This paper presents a brief overview of the fundamental design principles of the basic VQ structure and its variations. Deeper discussions of many of the issues and systems may be found in tutorial articles[4,5,6]. A thorough development of VQ systems may be found[7].

VQ is a form of "lossy" data compression in contrast to "lossless" data compression or noiseless coding. Noiseless codes are perfectly invertible and necessarily variable length codes. The most popular noiseless coding algorithms are Huffman coding, Rice codes, Lempel-Ziv Codes, and arithmetic codes[2,55-61]. These codes are in common use, especially for the compression of computer programs and data files which cannot tolerate errors. Noiseless compression is usually required in such situations where the compression system must be designed without any knowledge of the structure or end use of the original digital signal. On the other hand, any system which begins with an analog signal (such as microphone, camera, or analog sensor output) and produces a digital output is necessarily a lossy code as a continuous voltage cannot be reproduced perfectly from a digital representation. Given that all such analog-to-digital conversion systems are lossy, the goal of any such system is to provide the best quality (minimum loss) within the constraints of the system. As unpleasant as purposefully introducing distortion into a representation might sound to a user, it is preferable to the potential large insertion of uncontrolled distortion or the complete loss of the data caused by overwhelming available digital communication or storage capacity. In other words, if you are generating gigabits but the available communication channel only takes megabits, then you either compress to the best acceptable quality or you may get no useful data at all. VQ is an approach to minimizing the loss for a given communication or storage rate. It is based on the Shannon theory approach of defining and minimizing an objective distortion criterion for a given code rate. The minimization is accomplished using algorithms developed in communications, statistics, and cluster analysis. Next two sections summarize the basic approach.

MATHEMATICAL MODELS

The mathematical model for an analog-to-digital conversion system or for a data compression system is a source code subject to a fidelity criterion. A source {X(n);n=1,2,...} is a discrete time signal which in general is vector-valued. Let A denote the alphabet or collection of possible values of X(n)., for example, A may be k dimensional Euclidean vector space. For convenience we assume that the signal is a statistically "nice" process (e.g., the law of large numbers holds). The basic results extend to more general processes.

A code in the general sense is a mapping of the input sequence {X(n)} into a binary sequence (the encoder) together with a mapping of the binary sequence into a reproduction sequence {Y(n)} (the decoder). (We assume the encoded sequence is binary for convenience, in general it need only be from a finite alphabet.) The rate (or resolution) R of the code is the number of bits or binary symbols transmitted or stored per source symbol. In general this rate can be fractional, although in some examples it is useful to focus on integral values. A block source code is a code where each input block or vector (X(lk), X(lk+1),...,X((l+1)k-1)) is mapped into its binary code word in a way that does not depend on past or future actions of the encoder. The decoder is required to act in a similar fashion independently of past and future vectors. A block source code is also called a (memoryless) block quantizer or vector quantizer. The qualifier "memoryless" reflects the fact that such codes operate on vectors in a memoryless fashion (although the clearly have memory with respect to the individual symbols within the vectors). We will consider codes that have memory, but the focus of Shannon theory is on memoryless codes.

To measure the performance of a code, we assume a non-negative distortion measure d(x,y) which measures the cost of reproducing any x in A by some y in a reproduction alphabet B, which may or may not be the same as A. The performance is measured by an average distortion, where the average may be a long term time average or an ensemble or probabilistic average. For convenience we represent both by

$$\Delta_N = E\left[\frac{1}{N}\sum_{i=1}^{N} d(X(i), Y(i))\right],$$

where the expectation E can mean either a probabilistic average or a time average (which can be viewed as a special case of a probabilistic average in which every sample vector in a training sequence of length L has probability 1/L. Ideally a distortion measure should be analytically tractable, computable, and subjectively meaningful. In practice, these attributes must be balanced and a variety of choices exist.

Shannon's converse coding theorem and its generalizations imply that for any code for which these definitions make sense, Δ can be no smaller than the distortion-rate function D(R) of the source and distortion measure evaluated at rate R, a function defined by an information theoretic minimization which can be computed analytically or numerically or bounded for many interesting sources. Shannon also showed that performance arbitrarily near to D(R) could be achieved with block source codes, another name for VQ. Unfortunately, however, the proof of this result provided no indication of how to actually design a good code. This we explore shortly.

MEMORYLESS VECTOR QUANTIZATION

As described in the previous section, a memoryless vector quantizer or block quantizer is in the Shannon terminology a length k block source code subject to a fidelity criterion; that is, it is a pair of mappings, an encoder E which maps k-dimensional input vectors X into binary vectors which we denote by their equivalent decimal representation j = 1,2,...,M, and a decoder D which maps those binary vectors into reproduction vectors. For simplicity we assume that the binary vectors have dimension K and hence that the rate of the code is R = K/k bits per input symbol. To describe the operation of a block code define the code book C= {y (j) = D(j), j = 1,2,...,M} as the collection of all possible reproduction vectors, and the code partition P = { P(j);j=1,...,M}, where P(j) is the collection of all input vectors which are encoded into the binary vector j. The quantizer mapping Q(x)is defined as D(E(x)), that is, the overall action of the code. The term VQ is used to refer to combination of the encoder and decoder or, equivalently, the overall mapping Q.

The average distortion resulting from applying a vector quantizer to a source can be written using conditional averages as

$$\Delta = \sum_{j=1}^{M} E \quad [d(X,Y(j))|X\epsilon P(j)]P(X\epsilon P(j))$$

Recall that the expectation and the probabilities may come either from a probabilistic model or (more commonly) from a training sequence of typical data. A vector quantizer is optimal if it yields the smallest possible Δ over any quantizer with the same dimension and resolution. The above representation easily yields two necessary conditions for a VQ to be optimal.

The Nearest Neighbor (Minimum Distortion) Condition

A necessary condition for a VQ to be optimal is that the encoder be optimal for the decoder. This is equivalent to the following: If the decoder yields a code book C, then the encoder must be a nearest neighbor or minimum distortion mapping that satisfies

$$E(x) = j \text{ only if } d(x,y(j)) \leq d(x,y(i)), \text{ all } i \neq j$$

Thus given a decoder or, equivalently, a code book, the optimal encoder is the one which searches the entire code book and selects the binary vector corresponding to the minimum distortion available reproduction vector.

The Centroid Condition

Define the centroid of a set S with respect to a distortion measure d and a probability distribution on X = (X(1), ... ,X(N)) by

$$\text{cent}(s) = \min_{Y}^{-1} E[d(X,Y)|X\epsilon S]$$

A necessary condition for a VQ to be optimal is that the decoder be optimal for the encoder. This is equivalent to the following: If the encoder implies a partition {P(j)}, then the optimal decoder satisfies

$$D(j) = \text{cent}(\ P(j)); \quad \text{all } j.$$

For example, in the case of mean squared error, the centroid is simply a conditional mean given that the input was mapped into the binary vector j. This condition states that given an encoder, which can be considered as a partition of the input vector space, then the optimum decoder is the one which maps each received binary vector into the centroid of the region of the partition which is encoded into that binary vector. Note that unlike the encoder condition, this condition requires knowledge of the input signal distribution, knowledge that can come from a mathematical model or from a training sequence.

MEMORYLESS VQ DESIGN

Because of the nearest neighbor condition, a VQ is completely described by its code book. The two properties together provide a means of improving any given code book C:

The Lloyd Iteration

1. Given a code book C, form a nearest neighbor partition {P(j)}.

2. Given a partition {P(j)}, form a new code book C'= {cent(P(j)); j=1, ... , M}.

It is obvious that the above operation produces a new code book that is better than (at least no worse than) the original code book since each step can only improve performance. These properties were first observed for the mean squared error and scalar quantizers by Lukaszewicz and Steinhaus$^{(8)}$ and were independently found shortly thereafter by Lloyd$^{(9)}$, who developed an iterative algorithm for designing scalar quantizers with a mean squared error based on repeated use of the iteration. The basic idea extends immediately to vectors and is called the generalized Lloyd algorithm:

The Lloyd VQ Design Algorithm

0. Given an initial code book C(0) and a threshold ∂.
   Set $\Delta(0)$ = huge. Set k = 1.
1. Use the Lloyd iteration to produce a new code book C(k) from C(k-1). 2. Evaluate the distortion

$$\Delta_k = E(\min_Y d(X,Y))$$

If

$$\frac{\Delta_{k-1} - \Delta_k}{\Delta_k} \leq \partial$$

quit. Otherwise replace k by k+1 and continue.

In most practical applications, one does not have a probability distribution, but does have a training sequence of data. In this case the algorithm can be run on the empirical distribution which assigns a probability of 1/L to each of L samples in the training sequence. In this case the expected distortion is replaced by a sample average.

Theorems can be proved to the effect that if the training sequence is long enough, the VQ designed will be close to that which would have been designed had the distribution been known(10,11).

This algorithm was developed in the statistical literature under the name of the k-means algorithm by MacQueen(12) and was first applied to vector quantization in the two dimensional case with a mean squared error by Chen(13). The algorithm was extended to general vector quantization with a variety of distortion measures by Linde, Buzo, and Gray(14), who computed centroids for a variety of distortion measures and applied the algorithm to speech waveform and voice compression.

A remaining issue is how to design the initial code book, which is in itself a code design problem. We now next describe several such techniques. In fact, these techniques can be used as an alternative to the Lloyd algorithm for designing a complete code, but such code books can always be improved by subsequent application of the Lloyd algorithm.

Random Coding

Perhaps the simplest conceptual approach towards filling a code book of M code words is to randomly select the code words according to the source distribution, which can be viewed as a Monte Carlo code book design. The obvious variation when designing based on a training sequence is to simply select the first M training vectors as code words. If the data is highly correlated, it will likely produce a better code book if one takes, say, every Nth training vector. This technique has often been used in the pattern recognition literature and was used in the original development of the k-means technique by MacQueen(12). One can be somewhat more sophisticated and randomly generate a code book using not the input distribution, but the distribution which solves the optimization problem defining Shannon's distortion-rate function. In fact, the Shannon source coding theorems imply that such a random selection will on the average yield a good code. Unfortunately, the code book will have no useful structure and may turn out to be quite awful.

Observe that here "random coding" means only that the code book is selected at random: once selected it is used in the usual (nearest neighbor) deterministic way.

Pruning

A variation on the above use of a training sequence to populate a code is to form a code book recursively as follow: Put the first vector in the training sequence in the code book. Then compute the distortion between the next training vector and the first code word. If it is less than some threshold, continue. If it is greater than the threshold, add the new vector to the code book as a codeword. Continue in this fashion: With each new training vector, find the nearest neighbor in the code book. If the resulting distortion is not within some threshold, add the training vector to the code book. Continue in this fashion until the code book has enough words. This technique has been used in the statistical clustering literature(15).

Product Codes

In some cases a product code book may provide a good initial guess. For example, if one wishes to design a code book for a k-dimensional VQ with $M = 2^{kR}$ code words for some integer R, then one can use the product of k scalar quantizers with 2^R words each. Thus, if q(x) is a scalar quantizer, then Q(x(1), ... , x (k))= (q (x(0)), ... , q(x (k)), the cartesian product of the scalar quantizers, is a vector quantizer. This technique will not work if R is not an integer. In general other product structures can be used, e.g., one could first design a one dimensional quantizer q from scratch (perhaps using a uniform quantizer as an initial guess). One could then use (q(x (0) , q(x(1)) as an initial guess to design a good two-dimensional quantizer Q(x(0), x(1)). One could then initiate a three dimensional

VQ design with the product (q(x(0)), Q(x(1), x(2))) as an initial guess. One could continue in this way to construct higher dimensional quantizers until the final size is reached.

Splitting

Linde et al. introduced a technique that resembles the product code initialization in that it grows large code books from small ones, but differs in that it does not require an integral number of bits per symbol(14). The basic idea is this: The globally optimal rate 0 code book of a training sequence is the centroid of the entire sequence. The one code word, say w(0), in this code book can be "split" into two code words, w(0) and w(0)+∂, where ∂ is a vector of small Euclidean norm. This new code book has two words and can be no worse than the previous code book since it contains the previous code book. The Lloyd algorithm can be run on this code book to produce a good resolution 1 code. When complete, the training sequence can be divided into two smaller training sequences (one for each of the two binary code words). For each of these sub-training sequences and the corresponding single reproduction vector we can repeat the design process: each of the code words in the new code book is split, forming an initial guess for a rate 2 bit code book, and the Lloyd iteration is run to convergence, producing a binary code book for the corresponding sub-training sequence. One continues in this manner, using a good rate r code book to form an initial rate r+1 code book by splitting. This algorithm provides a complete design technique from scratch on a training sequence based on a training sequence. Another approach to splitting is an application of the "greedy" decision tree design(23): Instead of splitting every node in a given level or depth of the tree together, split one node at a time by only splitting that node contributing the largest distortion to the overall distortion. In other words, each time the "worst" node is split. If a particular branch reaches the final permitted depth of the tree, it is then no longer permitted to split. This technique was suggested for code design in(6) and again provides a means of building a code book from scratch. As we shall see, these designs provide a useful tree structure to the VQ that can be exploited for fast encoding. Yet another alternative is to begin with, say, a random code. Each new training vector is then associated with one of the M current code words and the training vector is added to a corresponding cluster or group. The code word for that group is then replaced by the centroid of all training vectors in the group, including the new addition. This is in fact the way the original k-means algorithm operated(12).

Pairwise Nearest Neighbor Merging

An alternative scheme for producing an initial guess is to begin by considering every member of the training sequence as a cluster. At each step one chooses a pair of two clusters and merges them by grouping the vectors in each in a new common cluster and assigning the new cluster its centroid as a code word. One can choose the pair that provides the best possible change in average distortion or approximately so(63,64).

We close this section by observing that in place of the Lloyd algorithm, one can also iteratively design VQ code books by using standard gradient search algorithms(16).

CONSTRAINED MEMORYLESS QUANTIZERS

A serious problem with an ordinary VQ is that its complexity and memory grow exponentially with resolution. Although the general structure of such searches is amenable to implementation using VLSI systolic arrays(17,18), any reduction in search complexity permits better performance at a given resolution. In the special case of mean squared error fidelity criteria, there are a variety of tricks that can be used to reduce the complexity of full searching of arbitrary code books(19). More generally, however, the easiest means of reducing search complexity is to impose additional structure on the code book in order to permit rapid searches for nearest neighbors or almost-nearest neighbors. The resulting loss of optimality may be compensated by the reduced implementation complexity. This can provide higher quality for a given rate and complexity, e.g., by permitting larger vector dimensions. Variations of the Lloyd algorithm can be run in order to produce a good code having the desired structure. Some of the code structures that have been studied are mentioned below:

Tree-Searched VQ

In a tree-searched VQ the full search of available codewords is replaced by a suboptimal tree search. The codeword is selected by a sequence of binary decisions instead of a single large search. (20,54,21,22,19,63,64). The advantage is that the complexity of the search will be linear in rate instead of exponential in rate. A disadvantage is that general algorithms providing good suboptimal searches for general codebooks are not known. A classical problem in computer science is to find fast nearest neighbor tree searches for unstructured code books. If, however, the code book is constructed with an eventual tree search in mind (a freedom not always possible in the computer science applications), then good design algorithms do exist, as considered next.

Tree-Structured VQ

A tree-structured vector quantizer (TSVQ) is an example of a tree-searched code where the codebook itself is forced to have a tree structure and hence the tree search algorithm is natural. In particular, the encoder makes R binary distortion computations and comparisons instead of a single search requiring $2^{(NR)}$ distortion computations. The first tree-search and tree-structured codes were designed by a variation of the splitting algorithm of VQ design: Use the splitting algorithm to design a complete code book and do not run the Lloyd algorithm on the final complete code book. Instead retain the entire tree used in the design, that is, all of the binary code books and the order in which they are used. This provides the TSVQ which is encoded by finding the minimum distortion path through the R layers of the tree; that is, one makes a sequence of R minimum distortion comparisons using a sequence of binary codes. After the final selection one achieves a terminal node (leaf) of the tree, which corresponds to the final code word. The decoder need only have the final code book and is a table lookup as before. A TSVQ is suboptimal because of its constrained structure, but it has two important advantages: It has low complexity and a fast encoder in comparison with an ordinary VQ since it needs to make only R binary comparisons instead of 2^R comparisons for a code book of size 2^R. In addition, the code has a nice successive approximation property in that each additional bit in the code word provides increased fidelity of the reproduction. This is a useful property, for example, in systems where the rate may be changed due to available communication channel capacity or where the communication link is slow and it is useful to get ever better quality reproduction as additional bits arrive.

Multistep VQ

Multistep VQ is a special case of tree-searched VQ where the same code book is used at all nodes at a given depth of the tree. This is usually accomplished by coding an error or residual produced by encoding the original signal using the code books previously encountered in the tree. For example, the first level does a coarse quantization of the input vector. The second level quantizes the difference of the original vector and the first level reproduction. The second level reproduction is formed by adding the first level reproduction to the second level error reproduction. The third level then quantizes the error resulting in the second level reproduction, and so on(24).

Hierarchical VQ

A VQ encoder can be constructed with a hierarchical structure, either by using long term parameter estimates to choose short term code books(25) or by using a sequence of ever longer dimension table lookups with a fixed code book size(26).

Product Codes

As previously described, a VQ code book can be decomposed into a cartesian product of smaller code books, a typical decomposition being into code books for gain (energy, mean, residual energy) and shape. Separate attributes are coded separately, but the coding is interdependent because the selection of specific code books and the distortion measure can depend on previously chosen code words(20,27,28).

Transform VQ

As a variation on a traditional scalar quantization technique, one can take a transform of a large window of data and then use VQ on the resulting transformed vectors(29). A similar (and older) technique is to filter the input process into subbands and use the vector generalization of subband coding by applying VQ to the separate outputs(30,31). By generalizing the notion of a transform to include any preprocessing of the data to enhance important features or well match human senses, good codes can often be found by combining linear filtering (possibly two dimensional) and VQ(32).

RECURSIVE QUANTIZERS

A VQ can be made to have memory by having a different code book for each state of the code. The encoder is given an input and a state and produces both an index of a code word in the state code book and a next state. The decoder tracks the state in order to decode using the correct code book. Two forms of recursive VQ have been extensively studied in recent years. The first is predictive VQ(33,16), a natural extension of predictive quantization (or DPCM) to vectors wherein a linear vector predictor is formed based on the decoded reproduction and subtracted from the input. The resulting residual is then put into a VQ. The second approach is finite state VQ (FSVQ), which is a form of switched VQ consisting of a finite collection of codebooks (each associated with an encoder state) together with a next-state rule which determines the next codebook from the current one and the encoded word. Finite state codes were introduced by Shannon(62) and design algorithms for vector quantization were developed in (34,35,36,37). Given the next-state rule, the design goal is to have an intelligent rule for selecting future code books based on past choices. These designs can be based on arguments from prediction theory, stochastic automata theory, or classification techniques. The latter approach, pioneered in (36,37,38), uses a simple classifier to detect important local image attributes such as background or edge detection and orientation. This classifier is used to divide a training sequence into vectors (blocks) which follow each occurrence of each class type. The sub-training sequences are then used to design memoryless VQs. An FSVQ is then constructed by classifying the decoder reproduction rather than the actual input, thus closing the loop and permitting the decoder to track the encoder state from a knowledge of the initial state and the received code words. These codes have produced excellent quality monochromatic images at 1/3 to 1/2 bit per pixel and real-time hardware implementations have been designed(39).

A general theoretical treatment of recursive VQ may be found in a recent book by Gabor and Gyorfi(40).

TRELLIS ENCODERS

If the decoder is a recursive VQ, then superior performance can be obtained by replacing the encoder by a trellis encoder (or delayed decision or multipath encoder). This produces a trellis encoding system which is designed using a variation of the Lloyd algorithm and such systems have proved quite effective in waveform coding applications(41,42,43). As with VQ, one can also design trellis codes by using gradient search techniques instead of the Lloyd algorithm(44,45)

MODEL VQ

Perhaps the most successful application of VQ to date has been that of LPC VQ, a form of model VQ wherein one codes an autoregressive (all-pole) model of a window of speech instead of the waveform itself(14,20). Here a complicated distortion measure such as the Itakura-Saito distortion is used, but the Lloyd algorithm still works easily since the distortion can still be written as an inner product and a simple centroid computed. By grouping several LPC vectors together, one can achieve even lower rates by similar techniques using matrix quantization(46,47,48). Model coders can be combined with waveform VQ to form a variety of hybrid and adaptive systems, e.g., (49,50,41,51,17).

VARIABLE RATE VQ

In many coding applications the activity of the data can vary widely over time. In such applications it is often useful to use variable rate codes, that is, codes that can use more bits for active vectors and fewer bits for dormant ones. The cost is in added buffering and software to ensure synchronization and possibly to meet a fixed rate communication channel requirement, but this cost is often justified by significant performance improvement. One approach is to simply combine VQ with traditional noiseless coding techniques. (See, e.g., (52).) Another approach is to design an inherently variable-rate VQ by using tree pruning algorithms from the theory of decision tree design to produce optimal variable-length subtrees from tree-structured vector quantizers (53). Here one which first designs a fixed rate tree-structured VQ using a Lloyd algorithm and then "prunes" the resulting tree using an extension of a decision tree design technique of Breiman, Friedman, Olshen, and Stone (23). By removing the leaves of a complete tree in an optimal fashion, one obtains a collection of codes with distortion-rate pairs that can strictly dominate even full searched unstructured VQ. The resulting system is simple in comparison with noiseless codes for such large alphabets and has the successive approximation property of TSVQ that each bit improves the fidelity. Hence such codes are well suited to applications where one wishes to improve the quality of an image (or a selected portion) as additional bits arrive. Traditionally such successive improvement has been accomplished by transform coding sending additional coefficients. The VQ approach has the potential for a smoother and locally optimal sequence of improvements.

WHERE NEXT?

The emphasis in VQ appears to have shifted from research to development, in particular to real-time VLSI implementations of speech and image coders at low and medium rates. In spite of this drift, several interesting possible directions for future research exist, among them being:

1. Fine tuning and comparing the many VQ variations with traditional transform techniques for a variety of data types, e.g., SAR, medical, video, and multispectral images.

2. Combined VQ and signal processing (e.g., transforming and windowing and the incorporation of models for human vision and voice into the signal processing and distortion measures) for the best possible quality compression at target bit rates.

3. Real time implementation using state-of-the art custom and

semi-custom chips.

4. Applications of VQ to speech and image recognition. Understanding the theoretical connections between clustering with minimum discrimination information distortion measures and Markov source modeling. Designing speech and image compression systems that are matched to subsequent processing by machine or human experts.

5. Using digital or analog associative or Hopfield memories for VQ implementation. Since VQ does not suffer much if a code word close to the nearest neighbor is selected instead of an exact nearest neighbor (unlike the case in error control coding), associative memories are well suited to this application.

ACKNOWLEDGEMENT

Much of the research described here was supported by the National Science Foundation and by ESL, Inc.

REFERENCES

1) Shannon, C.E., "A Mathematical Theory of Communication," Bell Systems Technical Journal, vol. 27, pp. 379-423,623-656, 1948.

2) Gallager, R.G., Information Theory and Reliable communication, John Wiley and Sons, New York 1968.

3) Berger, T., Rate Distortion Theory, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.

4) Gray, R.M., "Vector Quantization," IEEE ASSP Magazine, Vol. 1, pp. 4-29, April 1984.

5) Gersho, A. and V. Cuperman, "Vector Quantization: A pattern matching technique for speech coding," IEEE Communications Magazine, vol. 21, pp. 15-21, December 1983.

6) Makhoul, J., S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," Proceedings of the IEEE, vol. 73. No. 11, pp. 1551-1587, November 1985.

7) Gersho, A. and R. M. Gray, Signal Coding, Quantization, and Compression, 1987. In Preparation.

8) Lukaszewicz, J. and H. Steinhaus, "On Measuring by Comparison," Zastos. Mat., vol. 2, pp. 225-231, 1955. (In Polish.)

9) Lloyd, S.P., Least squares quantization in PCM. Unpublished Bell Laboratories Technical Note, 1957. Portions presented at the Institute of Mathematical Statistics Meeting Atlantic City New Jersey September 1957. Published in the March 1982 special issue on quantization of the IEEE Transactions on

Information Theory.

10) Gray, R.M., J. C. Kieffer, and Y. Linde, "Locally Optimal Block Quantizer Design," Information and Control, vol. 45, pp. 178-198, May 1980.

11) Sabin, M.J. and R. M. Gray, "Global Convergence and Empirical Consistency of the Generalized Lloyd Algorithm," IEEE Transactions on Information Theory, vol. IT-32, pp. 148-155, March 1986.

12) MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," Proc. of the Fifth Berkeley Symposium on Math. Stat. and Prob., vol. 1, pp. 281-296, 1967.

13) Chen, D.T.S., "On Two or More Dimensional Optimum Quantizers," Proceedings, 1977 International Conference on Acoustics, Speech, and Signal Processing, pp. 640-643, Hartford, CT, 1977.

14) Linde, Y., A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Transactions on Communications, vol. COM-28, pp. 84-95, January 1980.

15) Tou, J.T. and R. C. Gonzales, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.

16) Chang, P.G. and R. M. Gray, "Gradient Algorithms for Designing Predictive Vector Quantizers," IEEE Transactions on Acoustics Speech and Signal Processing, vol. ASSP-34, pp. 679-690, August 1986.

17) Davidson, G., M. Yong, and A. Gersho, "Real Time Vector Excitation Coding of Speech at 4800 bps," Proc. ICASSP, April 1987.

18) Davidson, G., P. Cappello, and A. Gersho, Systolic Architectures for Vector Quantization. Submitted for possible publication.

19) Gersho, A. and D. Cheng, "Fast Nearest Neighbor Search for Nonstructured Euclidean Codes," Abstracts of the 1983 IEEE International Symposium on Information Theory, p. 88, St. Jovite, Quebec, Canada, September 1983.

20) Buzo, A., A. H. Gray Jr. and R. M. Gray and J. D. Markel, "Speech Coding Based Upon Vector Quantization," IEEE Transactions on Acoustics Speech and Signal Processing, vol. ASSP-28, pp. 562-574, October 1980.

21) Gray, R.M. and H. Abut, "Full Search and Tree Searched Vector Quantization of Speech Waveforms," Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, pp. 593-596, Paris, May 1982.

22) Adoul, J.P., J. L. Debray and D. Dalle, "Spectral Distance Measure Applied to the Optimum Design of DPCM Coders with L Predictors," Proceedings of the 1980 IEEE International Conference on Acoustics Speech and Signal Processing, pp. 512-515, Denver Colorado, April 1980.

23) Breiman, L., J. H. Friedman, R. A. Olshen and C. J. Stone, Classification and Regression Trees, Wadsworth, Belmont, California, 1984.

24) Juang, B.H. and A. H. Gray Jr., "Multiple Stage Vector Quantization for Speech Coding," Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing, vol. 1, pp. 597-600, Paris, April 1982.

25) Gersho, A. and Y. Shoham, "Hierarchical Vector Quantization of Speech with Dynamic Codebook Allocation," Proceedings 1984 ICASSP, San Diego, March 1984.

26) Chang, P.C., J. May and R. M. Gray, "Hierarchical Vector Quantizers with Table-lookup Encoders," Proceedings 1985 IEEE International Conference on Communications, vol. 3, pp. 1452-1455, June 1985.

27) Baker, R.L. and R. M. Gray, "Differential Vector Quantization of Achromatic Imagery," Proceedings of the International Picture Coding Symposium, March 1983.

28) Sabin, M.J. and R. M. Gray, "Product Code Vector Quantizers for Waveform and Voice Coding," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-32, pp. 474-488, June 1984.

29) Chang, P.C., R. M. Gray and J. May, "Fourier Transform Vector Quantization for Speech Coding," IEEE Transactions on Communications, pp.1059-1068, 1987.

30) Heron, C.D., R. E. Crochiere and R. V. Cox, "A 32-Band Subband/Transform Coder Incorporating Vector Quantization for Dynamic Bit Allocation," Proceedings ICASSP, pp. 1276-1279, Boston, April 1983.

31) Kim, C.S., J. Bruder, M. J. T. Smith and R. M. Mersereau, "Subband Coding of Color Images using Finite State Vector Quantization," Proceedings ICASSP, pp. 753-756, 1988.

32) Budge, S.E., T. J. Stockham, Jr., D. M. Chabries and R. W. Christiansen, "Vector Quantization of Color Digital Images within a Human Visual Model," Proceedings ICASSP, pp. 816-819, 1988.

33) Cuperman, V. and A. Gersho, "Vector Predictive Coding of Speech at 16 Kb/s," IEEE Transactions on Communications, vol. COM-33, pp. 685-696, July 1985.

34) J. Foster, R. M. Gray, and M. O. Dunham, "Finite-state Vector Quantization for Waveform Coding," IEEE Transactions on Information Theory, vol. IT-31, pp. 348-359, May 1985.

35) Haoui, A. and D. G. Messerschmitt, "Predictive Vector Quantization," Proceedings ICASSP, vol. 1, pp. 10. 10. 1-10. 10. 4, San Diego, March 1984.

36) Ramamurthi, B. and A. Gersho, "Classified Vector Quantization of Images," IEEE Transactions on Communications, vol. COM-34, pp. 1105-1115, November 1986.

37) Aravind, R. and A. Gersho, "Image Compression Based on Vector Quantization with Finite Memory," Optical Engineering, vol. 26, pp. 570-580, July 1987.

38) Kim, Taejeong, "New Finite State Vector Quantizers for Images," Proceedings ICASSP, pp. 1180-1183, 1988.

39) Shen, H.-H. and R. L. Baker, "A Finite State/frame Difference Interpolative Vector Quantizer for Low Rate Image Sequence Coding," Proceedings ICASSP, pp. 1188-1191, 1988.

40) Gabor, G. and Z. Gyorfi, Recursive Source Coding, Springer-Verlag, New York, 1986.

41) Stewart, L. C., R. M. Gray and Y. Linde, "The Design of Trellis Waveform Coders," IEEE Transactions on Communications, vol. COM-30, pp. 702-710, April 1982.

42) Bei, C.D. and R. M. Gray, "Simulation of Vector Trellis Encoding Systems," IEEE Trans. on Communications, vol. COM-34, pp. 214-218, March 1986.

43) Ayanoglu, E. and R. M. Gray, "The Design of Predictive Trellis Waveform Coders Using the Generalized Lloyd Algorithm," IEEE Transactions on Communications, vol. COM-34, pp. 1073-1080, November 1986.

44) Freeman, G. H., "The Design of Time-invariant Trellis Source Codes," Abstracts of the 1983 IEEE International Symposium on Information Theory, pp. 42-43, St. Jovite, Quebec, Canada, September 1983.

45) Freeman, G. H., "Design and Analysis of Trellis Source Codes," Ph. D. Dissertation, University of Waterloo, Waterloo, Ontario, Canada, 1984.

46) Wong, D. Y., B. H. Juang and D. Y. Cheng, "Very Low Data Rate Speech Compression with LPC Vector and Matrix Quantization," Proceedings ICASSP, pp. 65-68, April 1983.

47) Tsao, C. and R. M. Gray, "Matrix Quantizer Design for LPC Speech Using the Generalized Lloyd Algorithm," IEEE Transactions on Acoustics Speech and Signal Processing, vol. ASSP-33, No. 3, pp. 537-545, June 1985.

48) Tsao, C. and R. M. Gray, "Shape-Gain Matrix Quantizers for LPC Speech," IEEE Transactions on Acoustics Speech and Signal Processing, vol. ASSP-34, No. 6, pp. 1427-1439, December 1986.

49) Rebolledo, G., R. M. G, "A Multirate Voice Digitizer Based Upon Vector Quantization," IEEE Transactions on Communications, vol. COM-30, pp. 721-727, April 1982.

50) Stewart, L. C., "Trellis Data Compression," Stanford Information Systems Lab Technical Report L905-1, July 1981.

51) Shroeder, M. R. and B. S. Atal, "Code-Excited Linear Prediction (CELP)," Proceedings ICASSP, Tampa, 198 pp. 1156-1159, 1988.

52) Ho, Y.S. and A. Gersho, "Variable-Rate Vector Quantization for Image Coding," Proceedings 1988 ICASSP, pp. 1156-1159, 1988.

53) Chou, P. A., T. Lookabaugh and R. M. Gray, "Optimal Pruning with Applications to Tree-structured Source Coding and Modeling," IEEE Transactions on Information Theory, to appear.

54) Gray, R. M., "Tree-searched Block Source Codes," Proceedings of the 1980 Allerton Conference, Allerton, Il., Oct. 1980.

55) Blahut, R. E., Principles and Practice of Information Theory, Addison-Wesley, Reading, Mass. 1987.

56) Witten, J. H., R. M. Neal and J. G. Cleary, "Arithmetic Coding for Data Compression," Communications of the ACM, pp. 520-540, Vol. 30.

57) Welch, T. A., "A Technique for High-performance Data Compression," Computer, pp.8-18, 1984.

58) Ziv, J. and A. Lempel, "Compression of Individual Sequences via Variable-rate Coding," IEEE Transactions on Information Theory, Vol. IT-24, 1978.

59) Rice, R. F. and J. R. Plaunt, "Adaptive Variable Length Coding for Efficient Compression of Spacecraft Television Data," IEEE Transactions on Commun. Tech., Vol. COM-19, pp. 889-897, 1971.

60) Rice, R. F., "Practical Universal Noiseless Coding," SPIE Symposium Proceedings, Vol. 207, San Diego, CA, August 1979.

61) Storer, J., Data Compression, Computer Science Press, 1988.

62) Shannon, C. E., Coding theorems for a discrete source with a fidelity criterion," IRE Natl. Conv. Record, Part 4, pp. 142-163, 1959.

63) Equitz, W., "Fast Algorithms for Vector Quantization Picture Coding," Proceedings ICASSP, pp. 18.1.1-18.1.4, April 1987.

64) Equitz, W., "Fast Algorithms for Vector Quantization Picture Coding," IEEE Transactions on ASSP, in review.

DATA COMPRESSION

CASE STUDIES

HAMPAPURAM RAMAPRIYAN

GODDARD SPACE FLIGHT CENTER

SESSION COORDINATOR

NOISELESS CODING, SOME PRACTICAL APPLICATIONS

Robert Rice
Jet Propulsion Laboratory

Abstract not available. Presentation covered material from references cited below which can be requested from the author.

REFERENCES

1) Rice, R.F., A.P. Schlutsmeyer, "Data Compression for NOAA Weather Satellite Systems," SPIE Seminar Proceedings, Vol. 249, July 1980.

2) Rice, R.F., "Block Adaptive Rate Controlled Image Data Compression," Proceedings 1979 National Telecom. Conf., Washington, D.C., Nov. 1979.

3) Rice, R.F., " End-to-End Imaging Information Rate Advantages of Various Alternative communication Systems," JPL Publication 82-61, JPL, Pasadena, CA., Sept. 1, 1982.

4) Rice, R.F., J. Lee, "Noiseless Coding for the Gamma Ray Spectrometer," JPL Publication 85-53, JPL, Pasadena, CA, June 1985.

5) Rice, R.F., J. Lee, "Noiseless Coding for the Magnetometer," JPL Publication 87-19, JPL, Pasadena, CA, June 15, 1987.

6) Rice, R.F., J. Lee, "Some Practical Universal Noiseless Coding Techniques Part II," JPL Publication 83-17, JPL, Pasadena, CA, March 1, 1983.

7) Rice, R.F., "Some Practical Universal Noiseless Coding Techniques," JPL Publication 79-22, JPL, Pasadena, CA, March 15, 1979.

8) Rice, R.F. "The Development of Efficient Coding for an Electronic Mail System," JPL Publication 83-64, JPL, Pasadena, CA, July 15, 1983.

N89-22341

A ROBUST COMPRESSION SYSTEM FOR LOW BIT RATE TELEMETRY - TEST RESULTS WITH LUNAR DATA

Khalid Sayood and Martin C. Rost
Department of Electrical Engineering
University of Nebraska

PROBLEM STATEMENT

The output of a Gamma Ray detector is quantized using a 14 bit A/D converter. The number of each of the 2^{14} or 16,384 levels occurring in a 30 second interval is counted. In effect, a histogram of the gamma ray events is obtained with 16,384 bins. The contents of these bins are to be encoded without distortion and transmitted at a rate less than or equal to 600 bits per second. Thus the contents of the 16,384 bins are to encoded using 18000 bits. The encoder should be simple to implement and require only a minimal amount of buffering.

PROPOSED SYSTEM

Encoder

The contents of the bins are treated as a sequence for purposes of encoding. The proposed system encoder can be divided into two stages (three if a Huffman coding option is used. See Figure 1.) The first stage is a leaky differencer whose input/output relationship is given by

$$z_n = x_n - [\ a x_{n-1}\ ]$$

where [t] is the largest integer less than or equal to t. The reason for using a leaky differencer is to allow the effect of errors to die out with time.

The output of the differencer forms the input for the second stage which is a modified runlength encoder. The encoder codebook contains six different types of symbols.

Mn - symbol used to represent negative differencer output values, for example, the differencer output values -1, -2,...,-n, are represented by the symbols M1, M2,...,Mn, respectively.

Pn - symbols used to represent positive differencer values, they are coded similar to the Mn symbols. Thus a differencer output value of +3 would be represented by the symbol P3.

Zn - symbols used to represent string of zeros of length n. Since the number of Z-symbols is kept small, these symbols represent "short" string of zeros (0-strings), while the S0- and S1-symbols to be introduced later represent "long" 0-strings.

BR - In the encoding scheme that follows, there will sometimes be a need to specify the end of a sequence. The BR or break symbol is used for this purpose.

SOXX - symbol used to represent long 0-strings. The S0 symbol indicates that a 0-string is being represented while X stands for a four bit word. XX is thus an eight bit word specifying the length of the 0-string.

S1XX - symbol used to represent long 0-strings that are followed by a 1. It is constructed in the same manner as the S0XX symbol.

Each symbol, Mn, Pn, Zn, BR, S0, and S1 is represented by a four bit word. The number of symbols in the encoder codebook is o(M)+o(P)+o(Z)+3 where o(M), o(P), and o(Z) are, respectively, the number of negative source symbols, positive source symbols, and short 0-strings symbols to be channel coded. As each symbol is represented by 4 bits, a total of sixteen encoder symbols are possible. In our coding scheme, o(M) is set to 2, o(Z) to 6, and o(P) to 5.

This means that if the differential output is -1, -2, 1, 2, 3, 4, 5 or a string of zeros of length five or less, it can be represented by a single symbol. What if the differential output is a positive value larger than five or a negative value less than -2? In such cases the largest (in magnitude) Mn or Pn symbol is used as a concatenation symbol. As an example, consider encoding the value 18.

Since o(P) is 5, the largest positive value that can be coded with a single symbol is 5. If P5 is also used as a concatenation symbol, larger source values can be coded. In this case, 18 can be coded as P5 P5 P5 P3. The receiver accumulates a total for all the P5 symbols consecutively received until a non-P5 symbol is received. This symbol is used to complete the current source value. In this case, P3 indicates the source value is 18.

In the case where the source value is a multiple of the maximum P-symbol value some confusion can occur in the decoding process. Consider the coding of the source values 10 followed by 8. In this case, four source symbols are required to code these values but, the receiver decodes them as a 18. To overcome this problem the break symbol (BR) is used. This symbol carries no data value but, is used by the receiver to prematurely stop the accumulation of P-symbols. Specifically, 10 and 8 are coded as P5 P5 BR P5 P3. The receiver stops constructing the first source value when the BR is encountered and start constructing the next with the following P5 symbol.

If a source value to coded is negative, the above procedure is used with the allowed M-symbols along with the BR symbol to prevent incorrect receiver decoding. For example, -3 would be encoded as M2M1 and -4 would be encoded as M2M2BR.

In this particular application, the tails of a given signal frame contain long runs of zeros that are separated by non-zero data values. It is very likely that these 0-string separators take the value 1. Thus, it is beneficial to code these runs with one of the following two symbols, each of which is three code words in length:

S0 x y a 0-string of length xy (base 16).

S1 x y a 0-string of length xy (base 16) followed a 1.

For example, the symbol, S0 4 0, represents a string of 64 0s, and the symbol, S1 4 0, represents a string of 64 0s followed by a 1. If the separating data value is not 1, then additional source symbols follow the S0 symbol to complete the description of its value. The maximum length of 0-string that can be coded with this type symbol is 255 (FF base 16). If a string of length greater than 255 is encountered, a concatenation rule must be applied.

Since the symbols S0 0 0 and S1 0 0 are not assigned, they are used as 0-string concatenation symbols. They are used to indicate the fact that a 0-string is to be built whose length is greater than 255. Each time one of these symbols is used it is assumed that a 0-string of length greater than 255 is being coded, and additional information is to be provided on its length by the following symbols. A 0-string is terminated if the last S0-symbol indicates a length value other than 00 for xy.

For example, if a 0-string of length 300 is followed by a 1, two source symbols (six channel words) are required to code the string: S1 0 0 S1 2 D. The value for xy of the first symbol is 00, so the 0-string is continued using the following S1-symbol(s). In this way, 0-

strings of arbitrary length can be constructed by concatenating as many S1 0 0 symbols as needed to bring the overall reconstructed 0-string length to within 255 0s of its full length. The final S1-symbol in such a series which does not have a 0 0 length indicator terminates the 0-string concatenation process. Since the S1 symbol is being used this 0-string is automatically followed by a 1. Consider coding a 0-string of length 300 that is followed by a -1. Two S0-symbols (six channel words) are required to code the 0-string, and one M-symbol (one channel word) is required to code the -1: S0 0 0 S0 2 D M1 for a total of seven channel words.

Since the long runlength symbols require three channel words each, an excessive amount of channel capacity can be wasted when coding short runs of 0s. As a consequence, a group of short run symbols that use only one channel word each are used to alleviate this problem. The identifier for these symbols is Zn (where n represents the length of the 0-string). For example, a run of 5 0s is represented by the symbol Z5. The coding length of a short 0-string using Zn symbols only improves the overall coding rate if the short 0-string is coded with fewer channel bits when using the Z-symbols instead of the S0- and S1-symbols.

Consider the following example for coding a string of 10 0s. Since o(Z) is 6, to code this 0-string using Z-symbols takes two channel words: Z6 Z4. But, when coded using an S0-symbol it takes three channel words to code this 0-string: S0 0 A. Therefore, the Z-symbol coding is more channel efficient. Since an S0- (or S1-) symbol always require three channel words, the only way to guarantee that short 0-strings are coded efficiently is to set the maximum number of short Z-symbols in a single 0-string coding to two. Thus, for an o(Z) of 6, the maximum 0-string length to be Z-symbol coded is 12.

The encoder described above has two main characteristics. First, it has been designed for the specific task noted in the problem statement. No claims are made regarding its suitability for other tasks. The second characteristic is its simplicity. The encoding operation requires a very small amount of computation. Furthermore, the onboard memory requirements for buffering are minimal.

If Huffman coding is to be used, the final stage of the encoder is a Huffman coder. This will, of course, increase the complexity of the encoder and may make the system more vulnerable to channel errors. Therefore, if at all possible we will avoid using a Huffman coder.

Decoder

The decoder for the proposed system consists of three stages. The first stage of proposed system decoder is maximum A Priori Probability (MAP) receiver[6]. The MAP receiver design is based on the assumption that the output of the encoder contains dependencies.

The MAP design criterion can be formally stated as follows: For a discrete memoryless channel (DMC), let the channel input alphabet be denoted by $A = \{a_0, a_1, ..., a_{M-1}\}$, and the channel input and output sequences by $Y = \{y_0, y_1, ..., y_{L-1}\}$ and $\hat{Y} = \{\hat{y}_0, \hat{y}_1, ..., \hat{y}_{L-1}\}$, respectively. If $A = \{A_i\}$ is the set of sequences $A_i = \{\alpha_{i,0}, \alpha_{i,1}, ..., \alpha_{i,L-1}\}, \alpha_{i,k} \epsilon A$, then the optimum receiver (in the sense of maximizing the probability of making a correct decision) maximizes P[C], where

$$P[C] = \sum_{A_i} P[C|\hat{Y}]P[\hat{Y}].$$

This in turn implies that the optimum receiver maximizes $P[C|\hat{Y}]$. When the receiver selects the output to be A_k, then $P[C|\hat{Y}] = P[Y = A_k|\hat{Y}]$. Thus, the optimum receiver selects the sequence A_k such that

$$P[Y = A_k|\hat{Y}] \geq P[Y = A_i|\hat{Y}] \ \nabla_i.$$

When the channel input sequence is independent, this simplifies to the standard MAP receiver[6]. Under conditions where this is not true, the receiver becomes a sequence estimator which maximizes the path metric. $\Sigma \log P(y_i|\hat{y}_i, y_{i-1})$(5). The path metric can be computed for a particular system by rewriting it using the following relationship(4).

$$P[y_i = a_j|\hat{y}_i = a_n, y_{i-1} = a_m] = \frac{P[\hat{y}_i = a_n|y_i = a_j]P[y_i = a_j|y_{i-1} = a_m]}{\Sigma_l\ P[y_i = a_l|y_{i-1} = a_m]P[\hat{y}_i = a_n|y_i = a_l]}$$

Notice that the right hand side consists of two sets of conditional probabilities $\{P[\hat{y}_i|y_i]\}$ and $\{P[y_i|y_{i-1}]\}$. The first set of conditional probabilities are the channel transition probabilities while the second depend only on the encoder output. The two are combined according to the above relationship to construct an M × M × M lookup table for use in decoding. The structure of the MAP receiver is that of the Viterbi decoder(4,5).

The second stage of the decoder is the inverse operation of the modified run-length encoder. The operation of this stage has already been described in the previous section. The final stage of the decoder is the inverse of the differential operation with an input output relationship

$$x_n = z_n + [ax_{n-1}]$$

RESULTS

In this section we present results obtained by using the proposed system of the previous section. The data used was provided by Ms.\ M. Mingarelli-Armbruster of the Goddard Space Flight Center. This data was generated according to a Poisson distribution where the Poisson parameter was obtained from ten hours of lunar data. Both noisy and noiseless channel performance of the proposed system were examined via Monte-Carlo simulation. A total of twenty, 30-second intervals were used in the tests. The performance was compared with the Rice algorithm(1-3).

Before proceeding with the results, some caveats are in order. First, the name Rice algorithm is a misnomer. What is presented$^{(1-3)}$ is not an algorithm but an approach. In this approach, a suite of algorithms is used to encode sections of the data, and the most efficient algorithm for that particular section of data is selected. In this way, data with very different statistical profiles can be accommodated. Thus what is presented$^{(1-3)}$ could more correctly be called the Rice Universal Coding Approach (RUCA). What we compare against here are algorithms presented$^{(1-3)}$ as examples of the RUCA. These algorithms were constructed for use in very general situations. As opposed to this, the particular algorithm presented here has been designed for a specific task. A final observation is that the encoder presented in this paper could very easily be used as the first stage of the RUCA. However, this would result in a rather complex encoder and substantial increase in the need for onboard memory over the proposed design. Therefore, if the algorithm presented in the previous section satisfies the requirements in terms of rate and robustness, such a step would be undesirable.

The results of the tests with both algorithms are presented in Table 1 and Table 2. The number of bits required to code twenty thirty-second intervals and the average rate needed for both algorithms is presented in Table 1. The second and third columns contain the total number of bits and the rate when the Rice algorithm is used. The average rate over twenty intervals is 719 bits per second. Columns three to six present the results obtained by using the proposed algorithm. The first two columns contain the results for the case where the Huffman coder was not used while the last two columns contain the results for when the Huffman coder formed the last stage of the encoder. The rate without the Huffman coder averaged over twenty intervals is 595 bits per second while the average rate when the Huffman coder is used is 522 bits per second. These results indicate that the proposed system will satisfy the specifications (coding rate below 600 bits per second) both when the Huffman coder is used and when it is not. As both systems meet the target and as the inclusion of the Huffman coder increases both the complexity and the vulnerability of the system to channel noise, we elected to use the system without the Huffman coder.

Table 2 provides the performance of the algorithms under noisy channel conditions. Three performance measures are used, namely, mean squared error (MSE), mean absolute error (MAE), and the number of decoded values which are in error. Note the very large difference between the performance of the Rice algorithm and the proposed algorithm. Also, the proposed algorithm maintains a robust performance at extremely high error rates. In fact, under even highly adverse conditions the mean squared error is almost constant, and the number of erroneous decoded values is about 25% of the total. However, the performance of the algorithms at high error rates may be irrelevant in this particular situation. The reason being that the transmitted data will be well protected by a channel coding scheme consisting of a Reed-Solomon coder followed by a convolutional coder. This combination is expected to keep the average probability of error on the coded channel below 9×10^{-6}.

Finally, we examine the relative complexity and buffer requirements for the two algorithms. The proposed algorithm can be easily realized with a simple program implemented using a microprocessor. Based on the memory requirements for the simulation program used in this study, the memory needed for actual implementation should be about 1 K. The only time búffering may be required is when a large differencer output is encountered, and the encoder has to generate several channel symbols for one input. Depending on the way the entire system is implemented, the buffer requirements could range from a single symbol buffer to perhaps a sixteen symbol buffer.

As opposed to this, the Rice algorithm by its very nature, being a universal coding algorithm, is quite complex. Each block of data is encoded using a number of candidate algorithms; the algorithm which provides the most efficient encoding is then selected. Each of the candidate algorithms is itself relatively complex though some very ingenious techniques are used to make subunits of one algorithm common to several candidate algorithms. Because several passes are required to do the encoding, the buffering requirements for this approach are substantial.

These differences in complexity are very natural based on the different objectives of the two algorithms. The proposed system is designed for a very specific situation while the Rice algorithm is designed to handle general situations.

TABLE 1

Coding rates with the Rice Algorithm and the Proposed Algorithm, (HC) denotes the results for the case where the Huffman Coder was used.

RICE ALGORITHM			PROPOSED ALGORITHM			
INTERVAL	TOTAL	RATE	TOTAL	RATE	TOTAL BITS	RATE
	BITS		BITS		(HC)	(HC)
1	21,647	721.6	17,832	594.4	15,733	524.4
2	21,385	712.8	17.528	584.3	15,345	511.5
3	21,530	717.7	17,784	592.8	15,520	517.3
4	21,562	718.7	17,840	594.7	15,691	523.0
5	21,666	722.2	18,144	604.8	15,883	529.4
6	21,424	714.1	17,504	583.5	15,457	515.2
7	21,841	728.0	18,048	601.6	15,882	529.4
8	21,630	721.0	18,096	603.2	15,907	530.2
9	21,719	723.9	18,132	604.4	15,843	528.1
10	21,568	718.9	18,096	603.2	15,695	523.2
11	21,308	710.3	17,604	586.8	15,438	514.6
12	21,509	716.9	17,728	590.9	15,580	519.3
13	21.633	721.1	17,780	592.7	15,581	519.4
14	21,822	727.4	18,016	600.5	15,913	530.4
15	21,296	709.8	17,564	585.4	15,361	512.0
16	21,701	723.4	17,956	598.5	15,872	529.1
17	21,058	701.9	17,296	576.5	15,139	504.6
18	21,312	710.4	17,688	589.6	15,449	514.9
19	21,713	723.8	18,160	605.3	16,033	534.4
20	21,888	729.6	18,292	609.7	16,125	537.5
OVERALL AVERAGE		718.7		595.1		522.4

TABLE 2

Performance of the algorithms under noisy channel conditions.

RICE ALGORITHM

PROBABILITY OF ERROR	MEAN SQUARED ERROR	MEAN ABSOLUTE ERROR	# OF DECODED ERRORS
10^{-6}	0.0760	0.023	140
10^{-5}	4.07	0.45	1,908
10^{-4}	31.49	3.14	10,177
10^{-3}	479.22	16.03	15,658
10^{-2}	8,562.87	76.75	16,189

PROPOSED ALGORITHM

PROBABILITY OF ERROR	MEAN SQUARED ERROR	MEAN ABSOLUTE ERROR	# OF DECODED ERROR
10^{-6}	2.4×10^{-5}	1.2×10^{-5}	1
10^{-5}	0.026	0.016	218
10^{-4}	0.17	0.14	1,287
10^{-3}	0.78	0.28	2,944
10^{-2}	6.81	0.71	3,765

SUMMARY AND CONCLUSIONS

We have presented a robust noiseless encoding scheme for encoding the gamma ray spectroscopy data. The encoding algorithm is simple to implement and has minimal buffering requirements. The decoder contains error correcting capability in the form of a MAP receiver. While the MAP receiver adds some complexity, this is limited to the decoder. Nothing additional is needed at the encoder side for its functioning.

ACKNOWLEDGMENT

This work was supported by a grant from the Goddard Space Flight Center, Greenbelt, Maryland, under Grant NAG5-916.

REFERENCES

1) R. F. Rice, "Practical Universal Noiseless Coding," 1979 SPIE Symposium Proceedings, Vol.\ 207, San Diego, CA, August 1979, pp. 247-267.

2) R. F. Rice, "Some Practical Universal Noiseless Coding Techniques," JPL Publication 79-22, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, March 15, 1979.

3) R. F. Rice and Jun-Ji Lee, "Some Practical Universal Noiseless Coding Techniques, Part II," JPL Publication 83-17, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, March 1, 1983.

4) K. Sayood and J. C. Borkenhagen, "Use of Residual Redundancy in the Design of Joint Source/Channel Coders," Submitted to IEEE Trans.\ Commun.

5) K. Sayood and J. D. Gibson, "Maximum Aposteriori Joint Source/Channel Coding," Proceedings of the 22nd Annual Conference on Information Sciences and Systems, Princeton, New Jersey, March 1988.

6) J. M. Wozencraft and I. M. Jacobs, "Principles of Communication Engineering. John Wiley and Sons, Inc., New York, 1965.

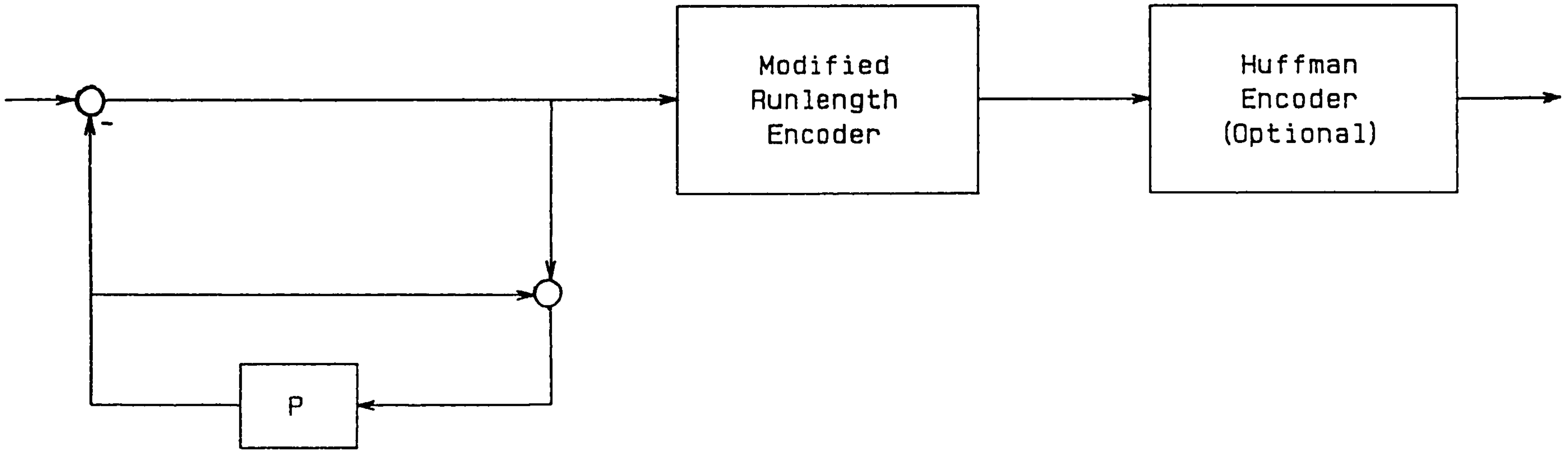


Figure 1. Proposed Encoder

THE OMV DATA COMPRESSION SYSTEM
SCIENCE DATA COMPRESSION WORKSHOP

Garton H. Lewis, Jr.
Fairchild Weston Systems

ORBITAL MANEUVERING VECHICLE (OMV)

COMPRESSION SYSTEM

PRESENTATION SUMMARY

* OVERVIEW VIDEO COMPRESSION UNIT – VCU

* OVERVIEW VIDEO RECONSTRUCTION UNIT – VRU

* THEORY AND ALGORITHMS FOR IMPLEMENTATION OF OMV SOURCE CODING

* DOCKING MODE – 2 – 510 X 488 CCD CAMERAS INTERLEAVED INTO 972 KBPS CHANNEL

* CHANNEL CODING – FORWARD ERROR DETECTION AND CORRECTION

* ERROR CONTAINMENT – RFI ENVIRONMENT

* VIDEO TAPE PROCESSED SPACE IMAGERY

PROGRAM OBJECTIVE

* PROVIDE REMOTE OMV PILOT WITH MONOCHROME VIDEO ADEQUATE FOR –

* TARGET VIEWING

* MANEUVERING OMV THROUGH FINAL 200 FT. DOCKING SEQUENCE

* DIGITAL COMPRESSION SYSTEM ALLOWS FOR FORWARD ERROR CORRECTION AND DETECTION

* OPERATION IN MODERATE RFI

DIGITAL COMPRESSION

* PROVIDE REDUNDANT CAMERA OUTPUTS IN ONE 972 KBPS – SSA CHANNEL

* PROVIDE FULL RESOLUTION MODE OVER 1/4 PIXELS –

* PROVIDE VERTICAL PIXEL PAIRED MODE

* PROVIDE FORWARD ERROR DETECTION AND CORRECTION – OPERATION MODERATE RFI

* ERROR CONTAINMENT – FOR ENTROPY ENCODING

* ENCRYPTION

ORIGINAL SOURCE CODING

- * BASED ON INADEQUATE SOURCE MATERIAL
- * BLACK RUN LENGTH CODING
- * SIMPLE DPCM
- * INEFFICIENT MARKER CODE
- * HAMMING CODE
- * ERROR CONTAINMENT – BASED ON FULL FRAME REPLACEMENT

UPDATED SOURCE CODING

- * APPROXIMATELY 130:1 COMPRESSION
- * 2–D FILTERING
- * 3–D DECIMATION
- * 2–D DPCM ADAPTIVE
- * ADAPTIVE PREDICTOR–LOCAL–GLOBAL
- * 16 NON – UNIFORM QUANTIZERS
- * DUAL–TIER VECTOR QUANTIZATION ERROR SIGNAL
- * HUFFMAN CODING
- * REED – SOLOMON EDAC
- * SUB – FRAME ERROR CONTAINMENT

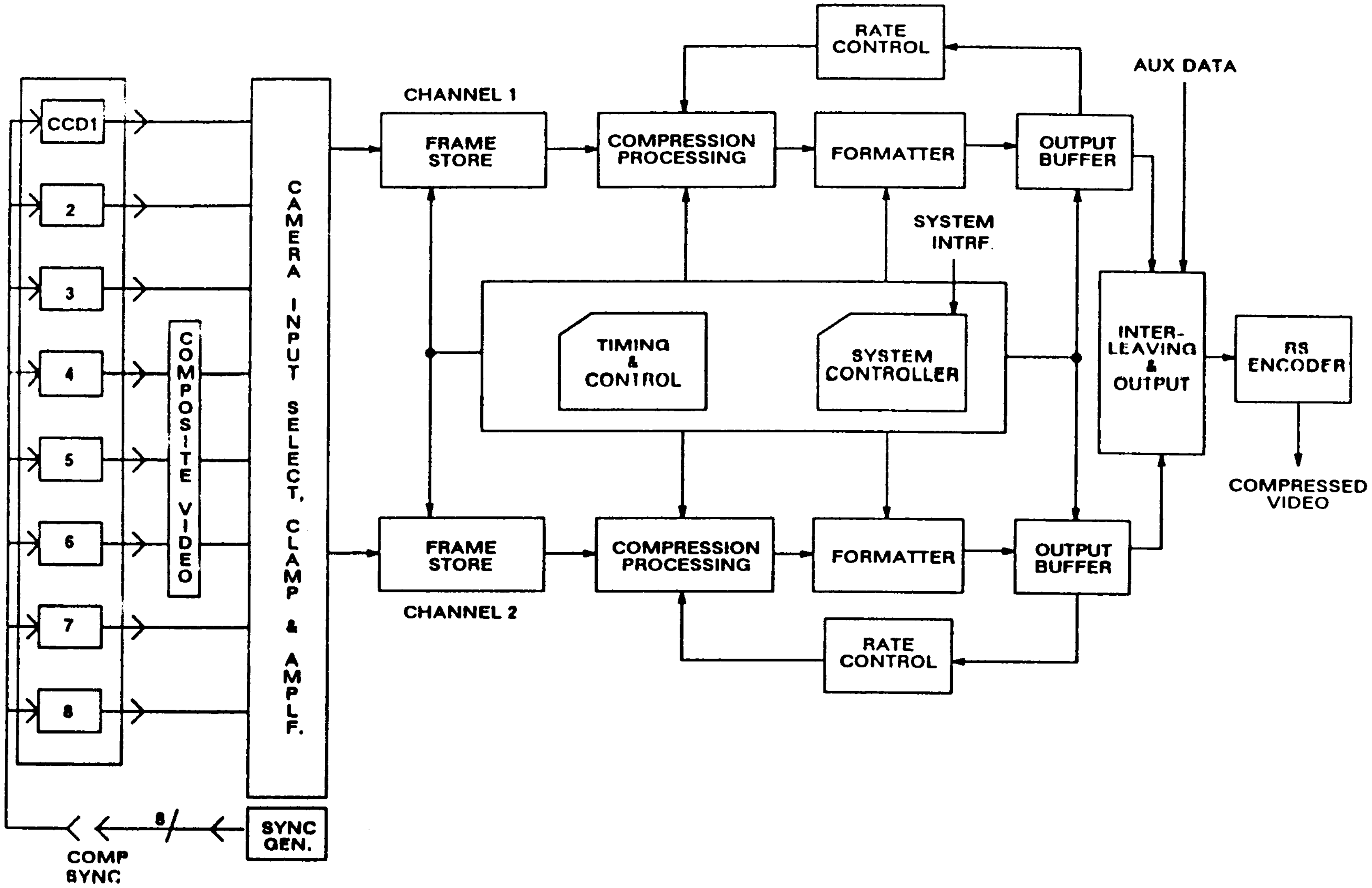
ALGORITHM CONSTRAINTS

1- LIMITED ACCEPTANCE OF VLSI-FLIGHT UNITS

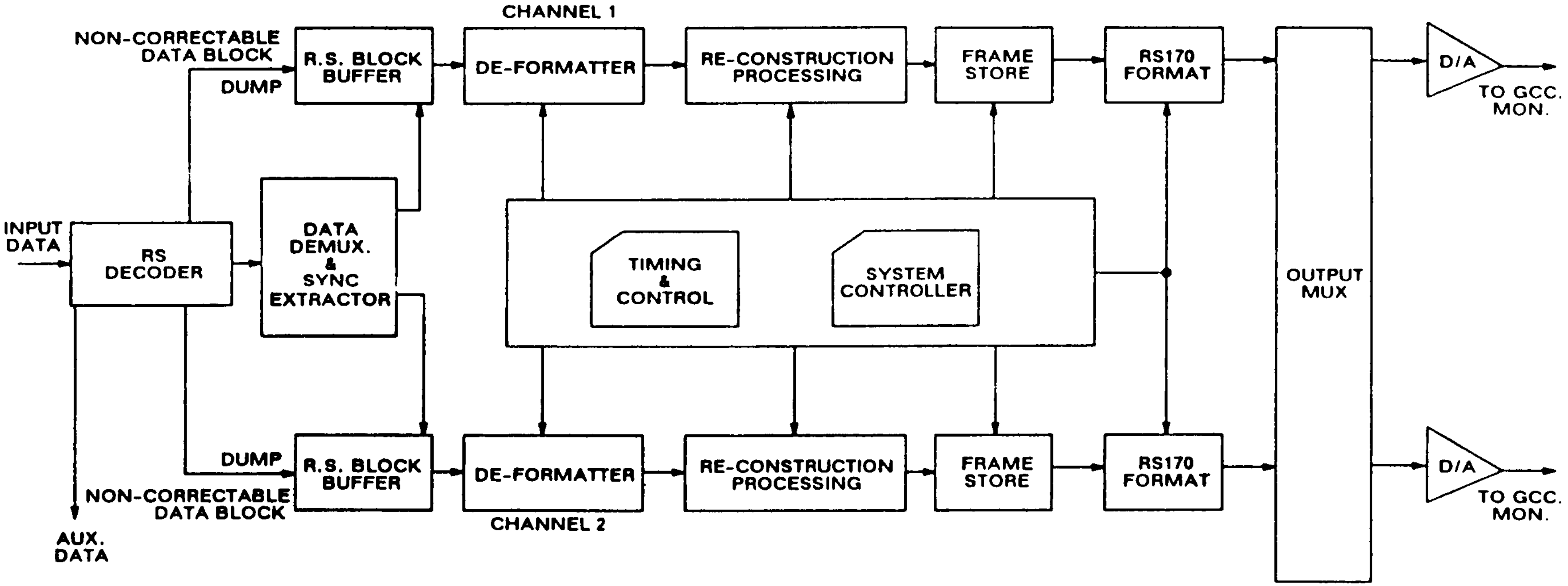
2- S-LEVEL PARTS

3- D.C. POWER BUDGET

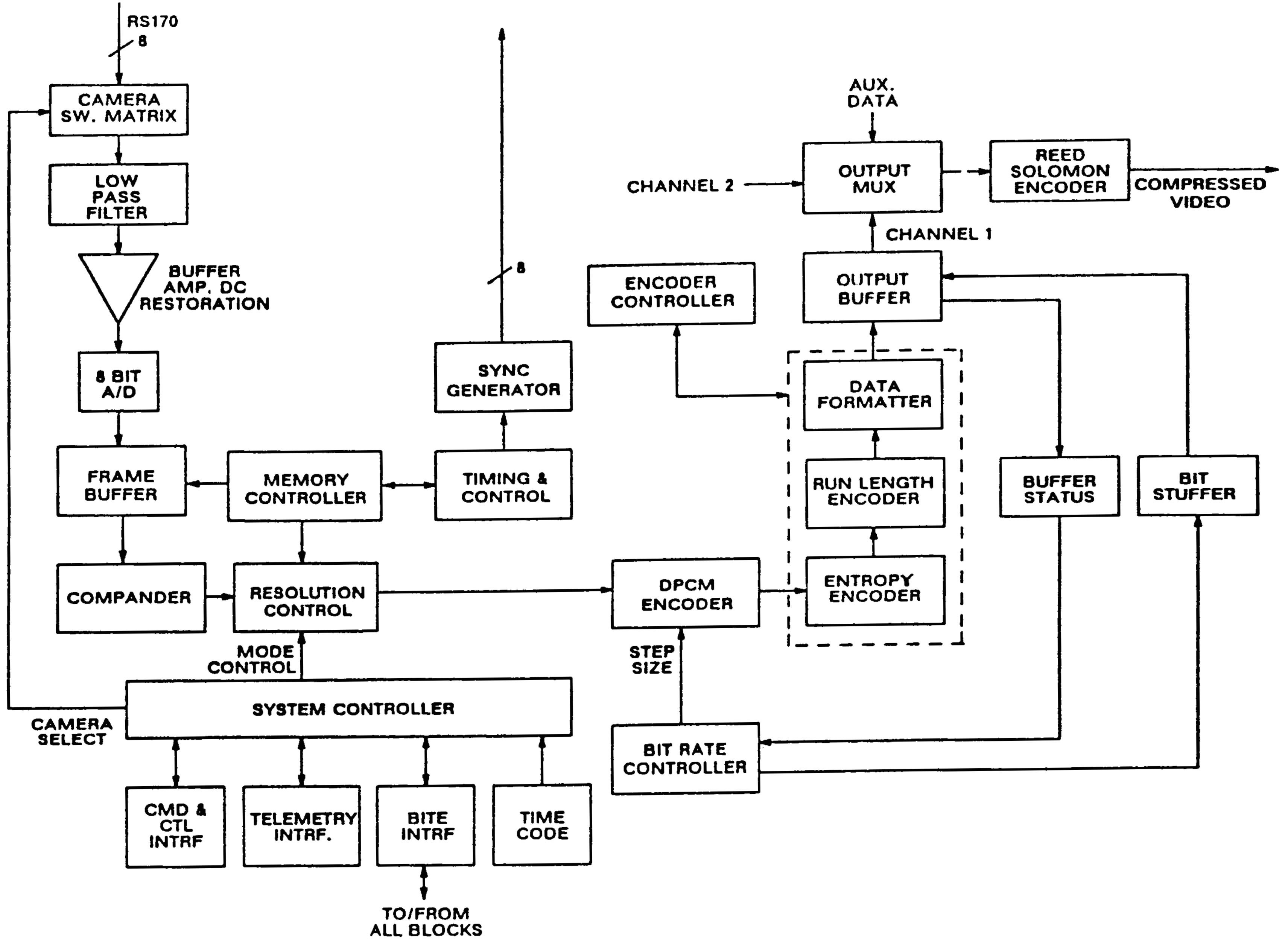
4- RFI CHANNEL ENVIRONMENT



VCU TOP-LEVEL BLOCK DIAGRAM



VRU TOP-LEVEL BLOCK DIAGRAM



VCU BASELINE FUNCTIONAL BLOCK DIAGRAM (1 Channel)

ARITHMETIC FORMAT

- HARDWARE IMPLEMENTATION 2'S COMPLEMENT FIXED POINT

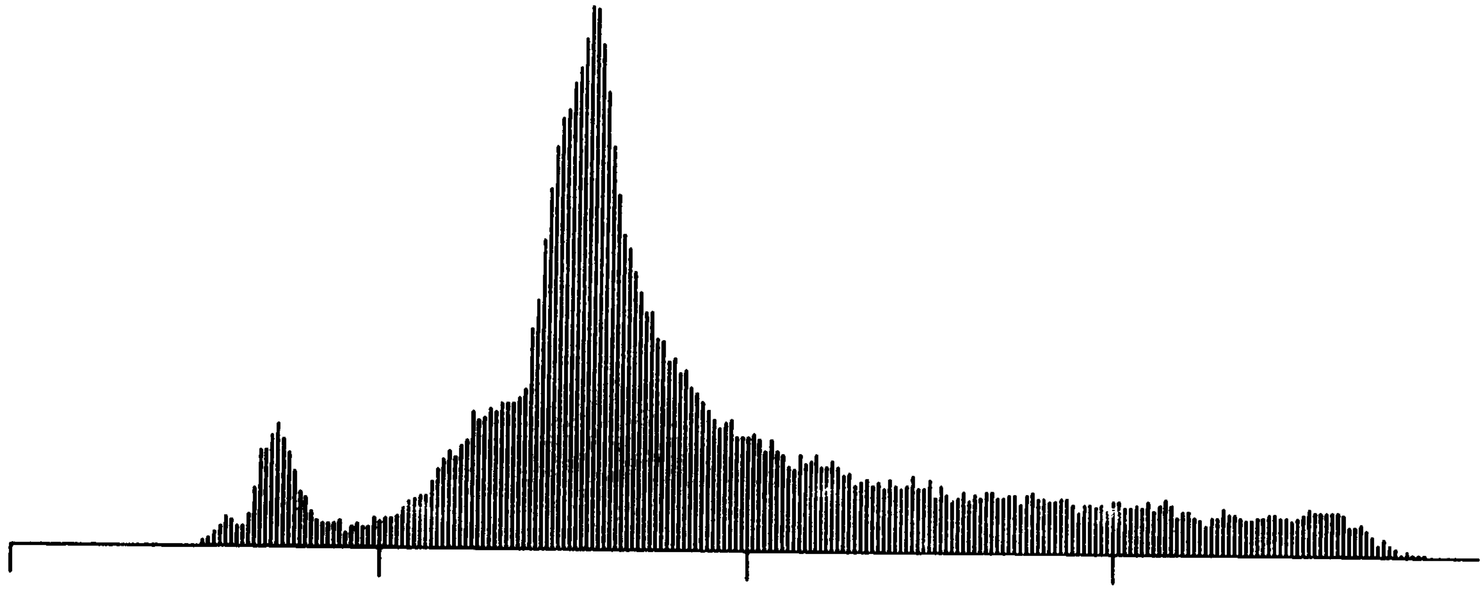
- FIXED POINT BINARY WHERE REQUIRED

- 2'S COMPLEMENT ROUNDING
  - POSITIVE NUMBERS ROUND UP
  - NEGATIVE NUMBERS TRUNCATE

- + 127 SIGNAL CLIPPED AND CLAMPED
  0
  – 127 TO EITHER +127 or –128 TO PREVENT OVERFLOW

OPERATION AT LOW LIGHT LEVELS

* INITIAL ACQUISITION AT 200 FT.
* 50 MM LENS
* LIGHT ON SUBJECT 0.03 FT. LAMBERTS
* LIGHT ON CCD 0.0015 FT. CANDLES
* INTEGRATE CCD CAMERA 1/5 SEC.
* INCREASE S/N IN ANALOG DOMAIN



BLACK SATELLITE: ORIGINAL IMAGE

2–DIMENSIONAL PIXEL PAIRING

LINE			510 X 488 ARRAY		
1	P1,1	P1,2	• • • • • •	P1,509	P1,510
2	P2,1	P2,2	• • • • • •	P2,509	P2,510
3	P3,1	P3,2	• • • • • •	P3,509	P3,510
4	P4,1	P4,2	• • • • • •	P4,509	P4,510
•	•	•		•	•
•	•	•		•	•
•	•	•		•	•
•	•	•		•	•
488	P488,1	P488,2	• • • • •	P488,509	P488,510

1* P*1,1 = 1/2 P1,1 + 1/2 P1,2 HORIZONTAL PIXEL PAIRING

2* P*2,1 = 1/2 P2,1 + 1/2 P2,2

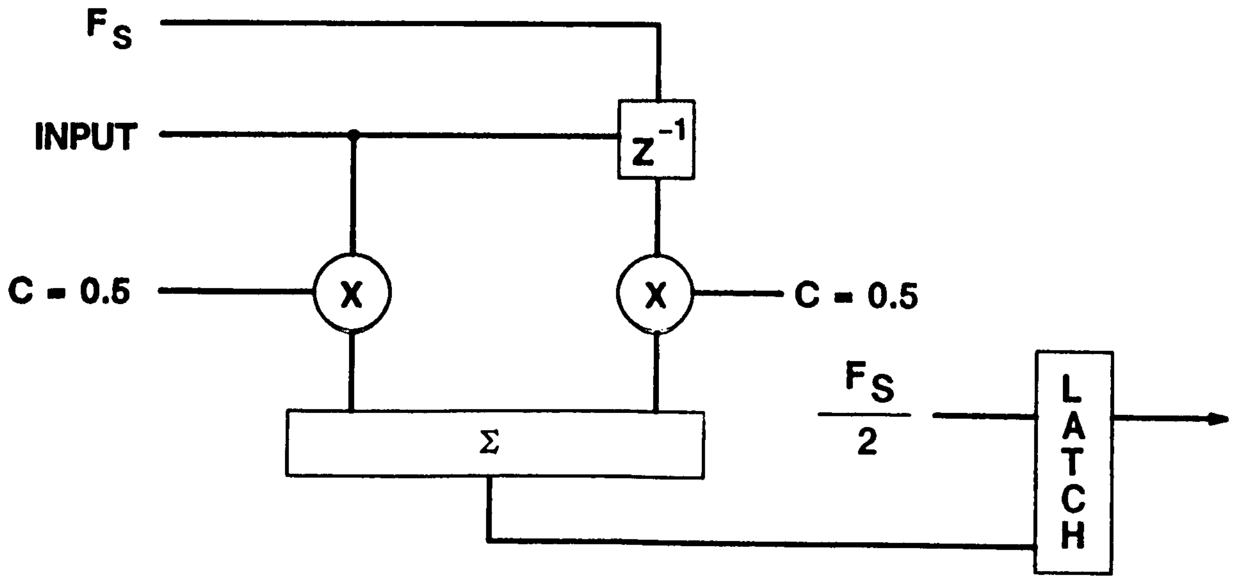
1** P**1,1 = 1/2 P*1,1 + 1/2 P*2,1 VERTICAL PIXEL PAIRING

PIXEL PAIRING AT 5 FRAMES / SECOND

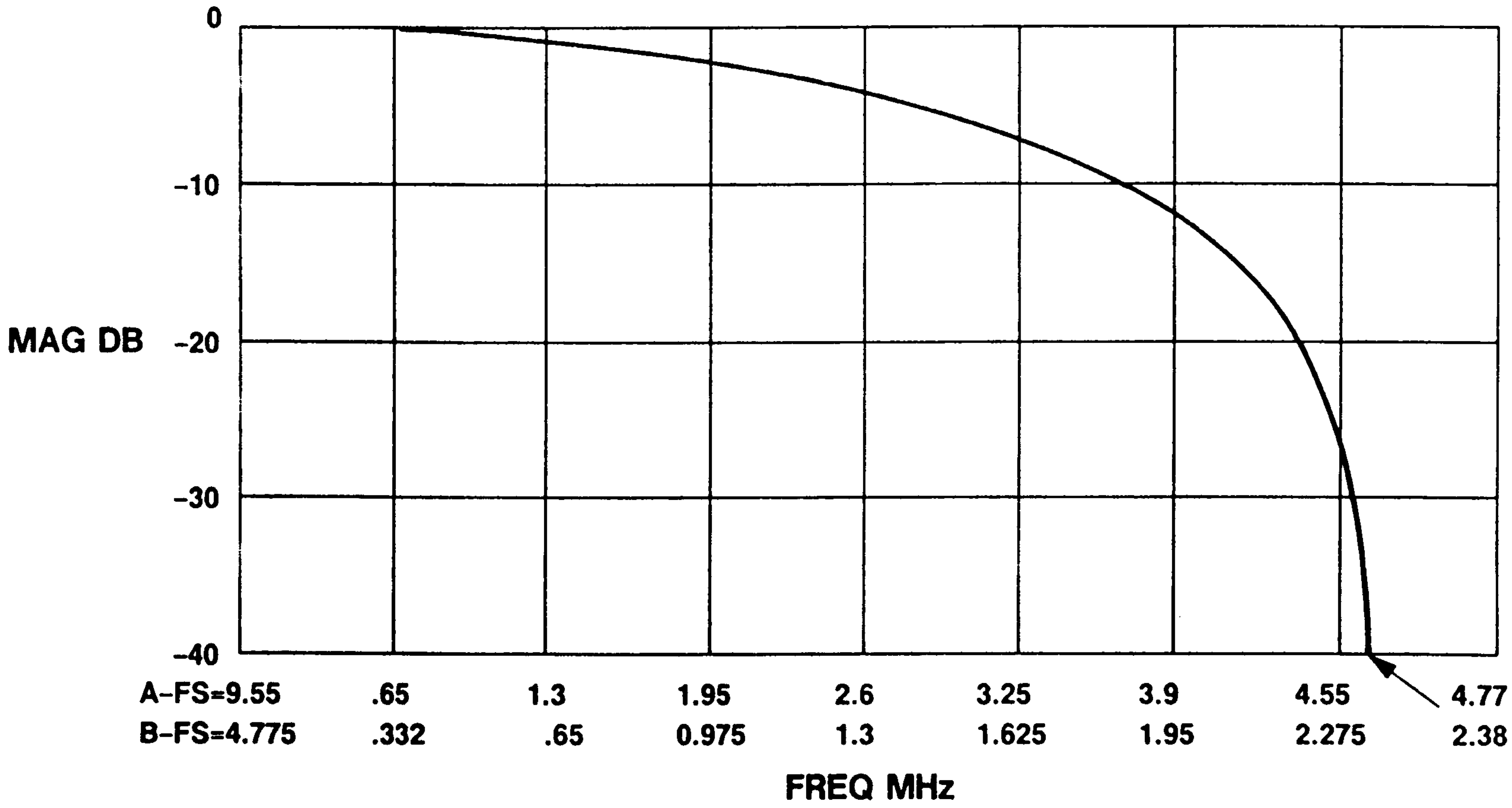
2 – DIMENSIONAL FILTERING RS–170 INPUT

HORIZONTAL $Hz = 0.5 + 0.5Z^{-1}$	DECIMATED 2:1
VERTICAL $Hz = 0.5 + 0.5Z^{-510}$	DECIMATED 2:1
TEMPORAL $Hz = Z^{-1}, Z^{-7}, Z^{-13}, Z^{-19}, Z^{-25}$	DECIMATED 6:1

FIR $Hz = 0.5 + 0.5Z^{-1}$



HORIZONTAL FILTER

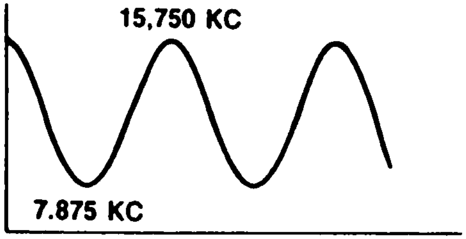


A – RESPONSE Hz = $0.5 + 0.5Z^{-1}$

B – RESPONSE Hz = $0.5 + 0.5Z^{-1}$ DECIMATED 2:1

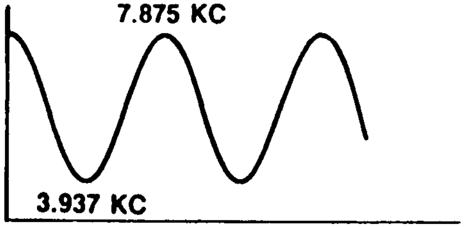
VERTICAL AND TEMPORAL FILTERS

$Hz = 0.5 + 0.5Z^{-H}$
H = LINE 607

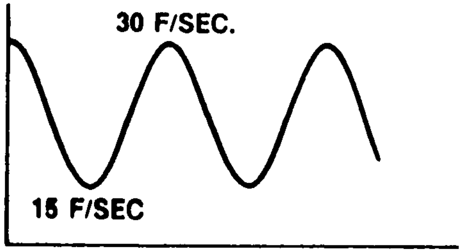


VERTICAL FILTER

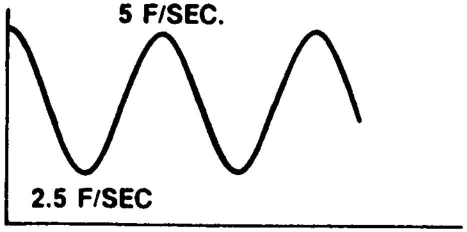
$Hz = 0.5 + 0.5Z^{-H}$
DECIMATED 2:1



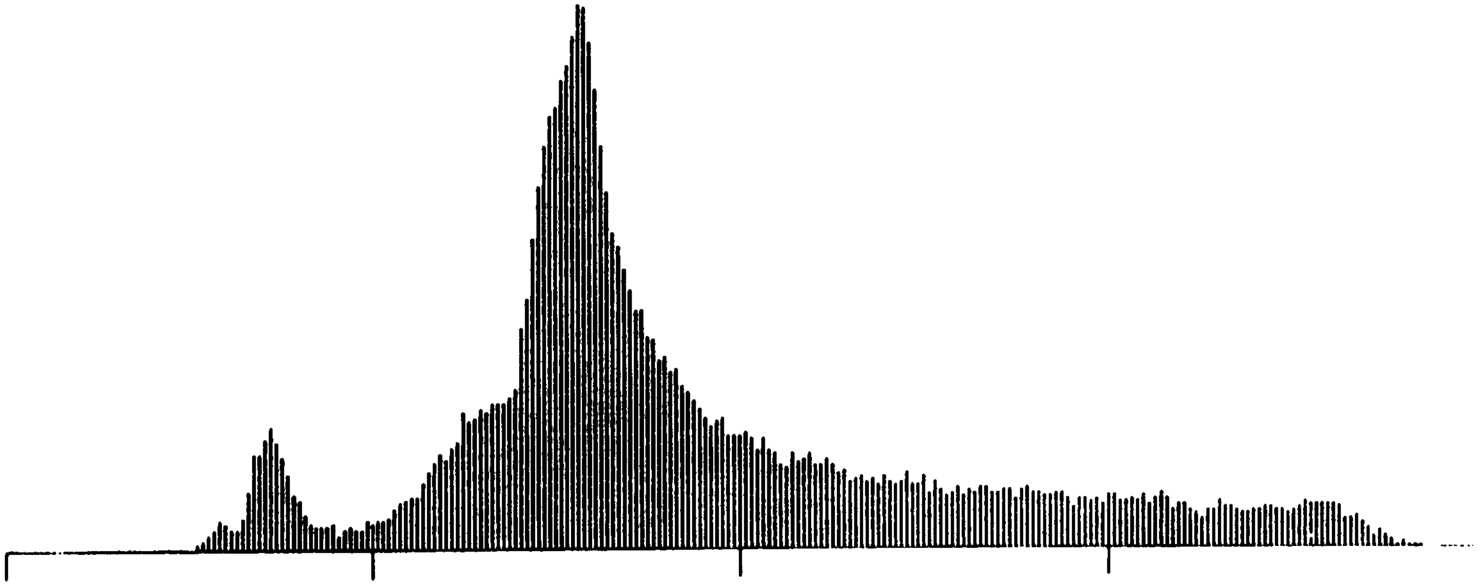
2:1 DECIMATION



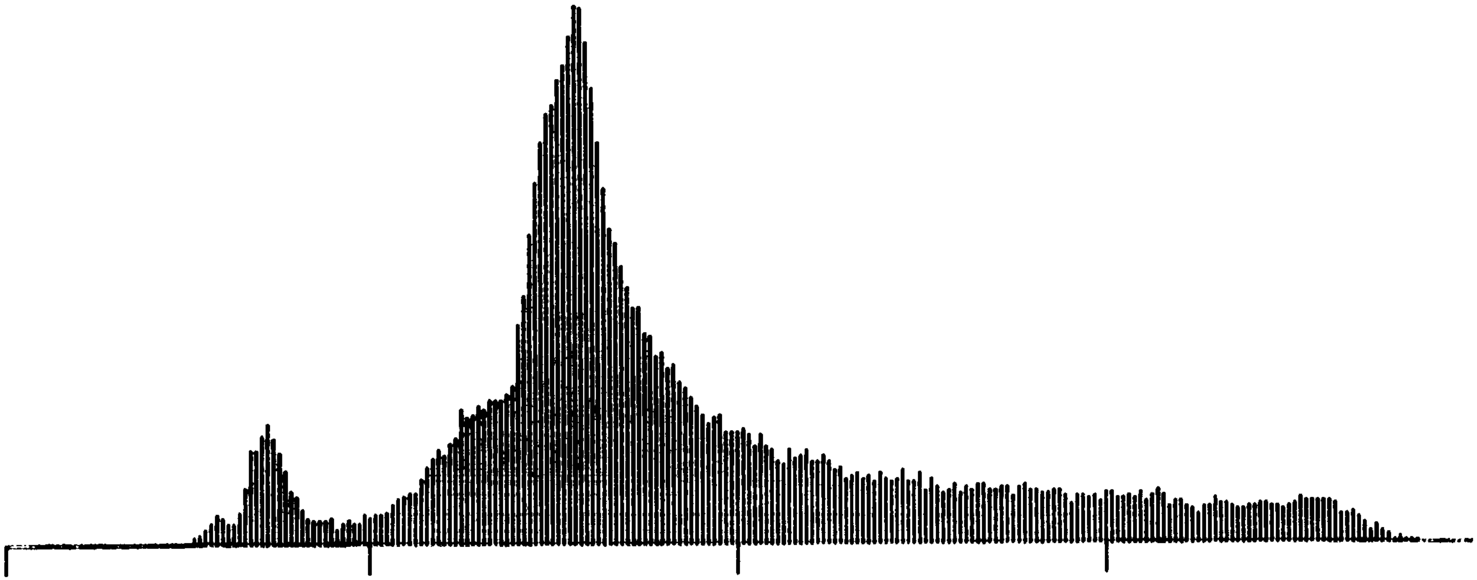
RS-170 TEMPORAL AXIS



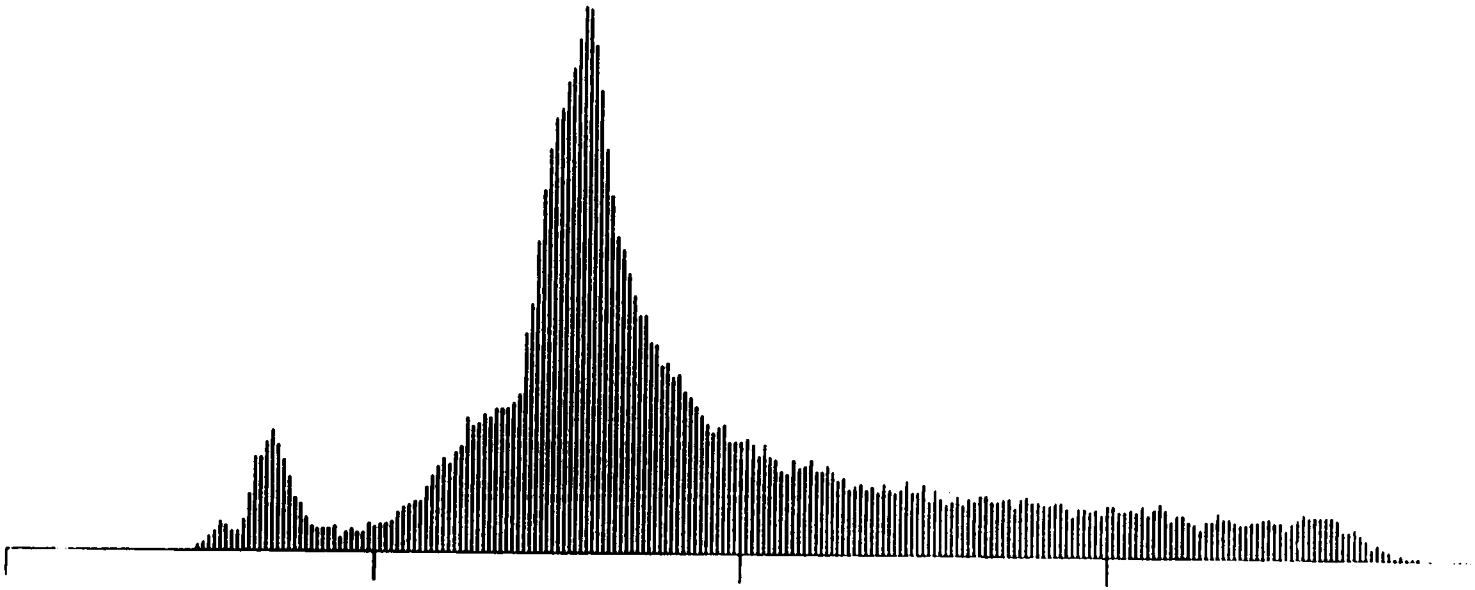
RS-170 6:1 DECIMATION



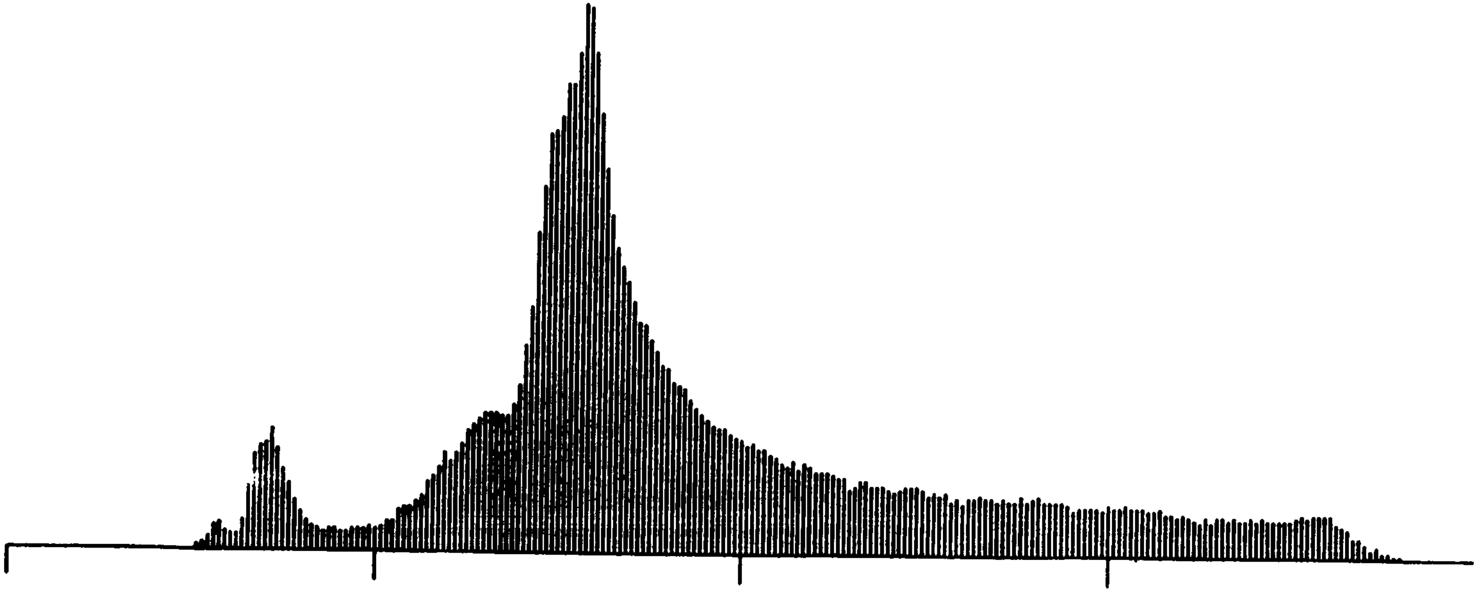
BLACK SATELLITE: ORIGINAL IMAGE



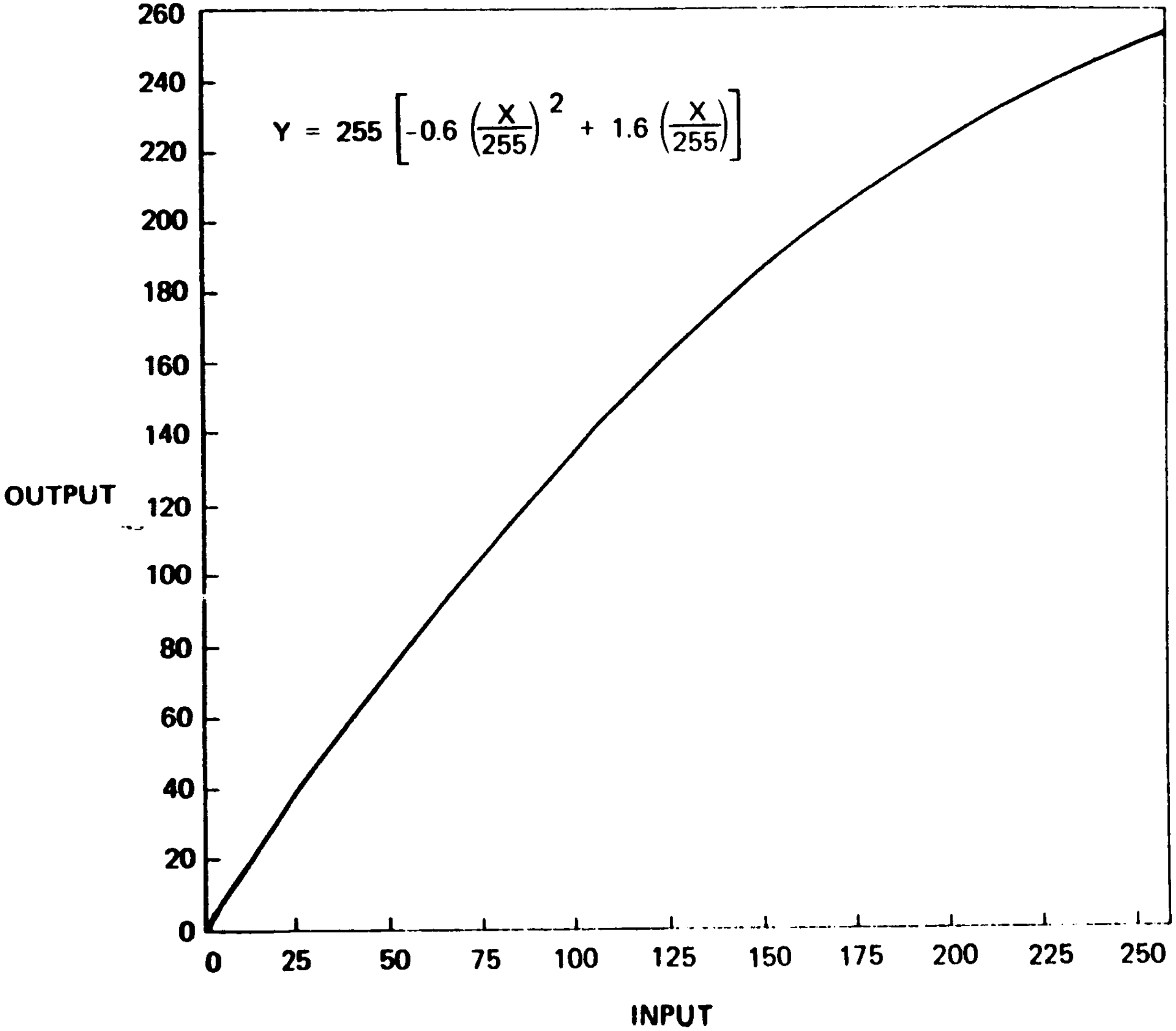
BLACK SATELLITE: EFFECT OF PIXEL PAIRING



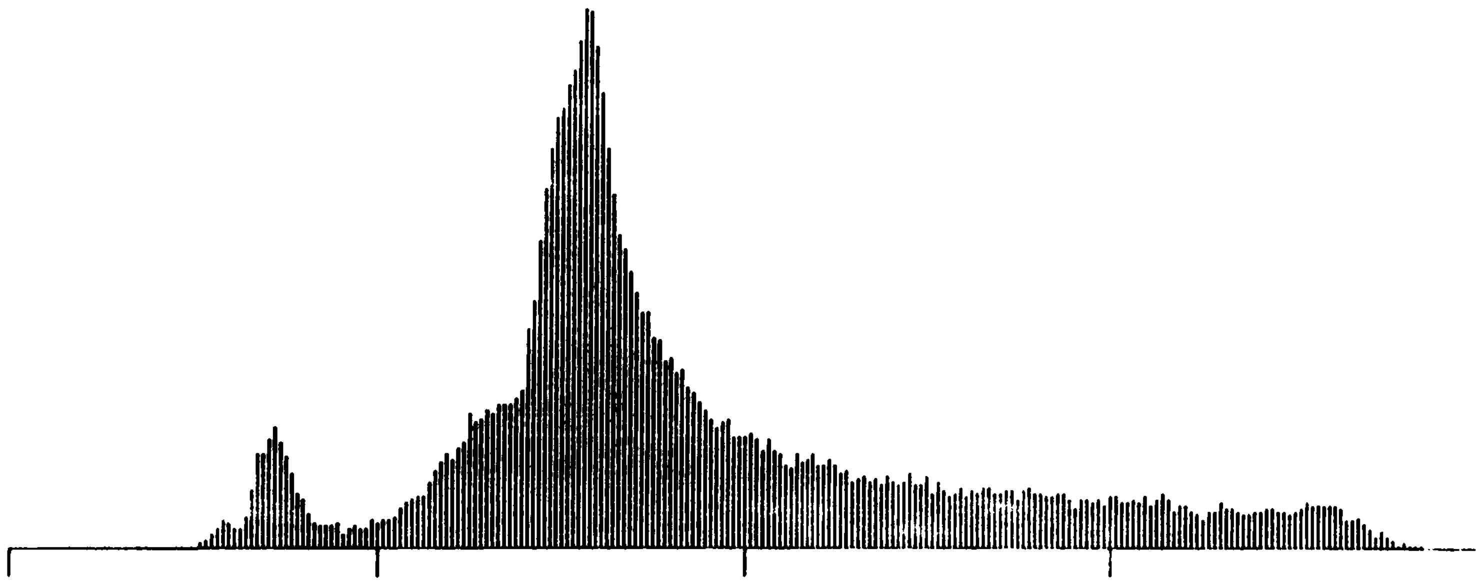
BLACK SATELLITE: ORIGINAL IMAGE



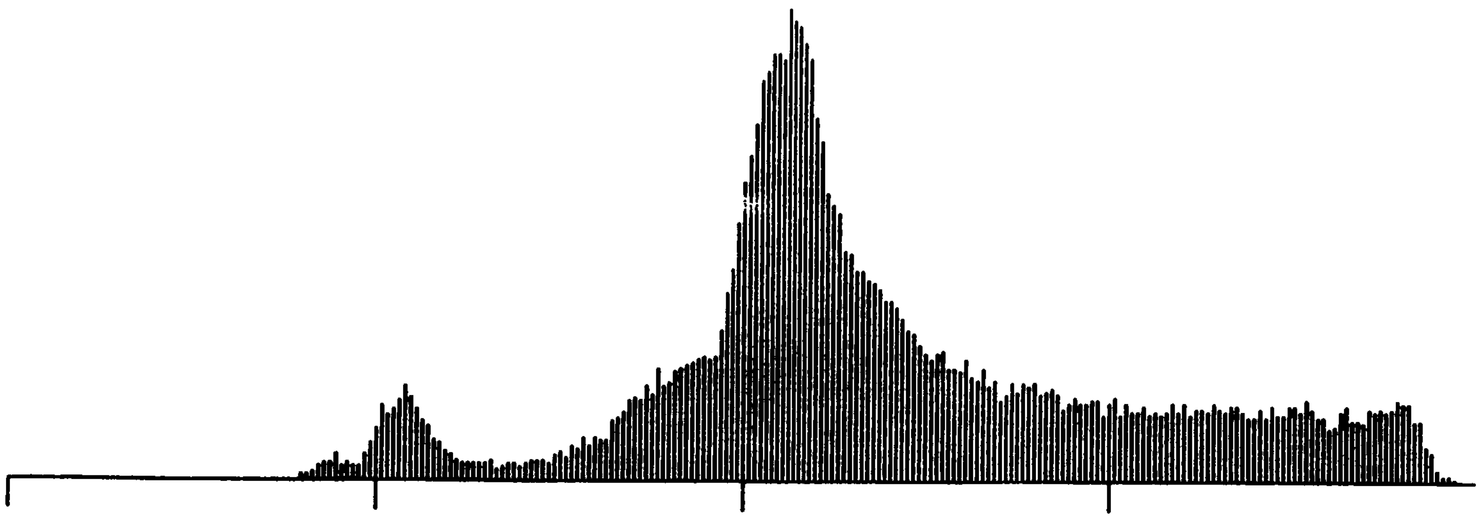
BLACK SATELLITE: PIXEL PAIRING AND INTERPOLATION


Y = 255 [-0.6 (X/255)^2 + 1.6 (X/255)]
OUTPUT
260
240
220
200
180
160
140
120
100
80
60
40
20
0
0
25
50
75
100
125
150
175
200
225
250
INPUT

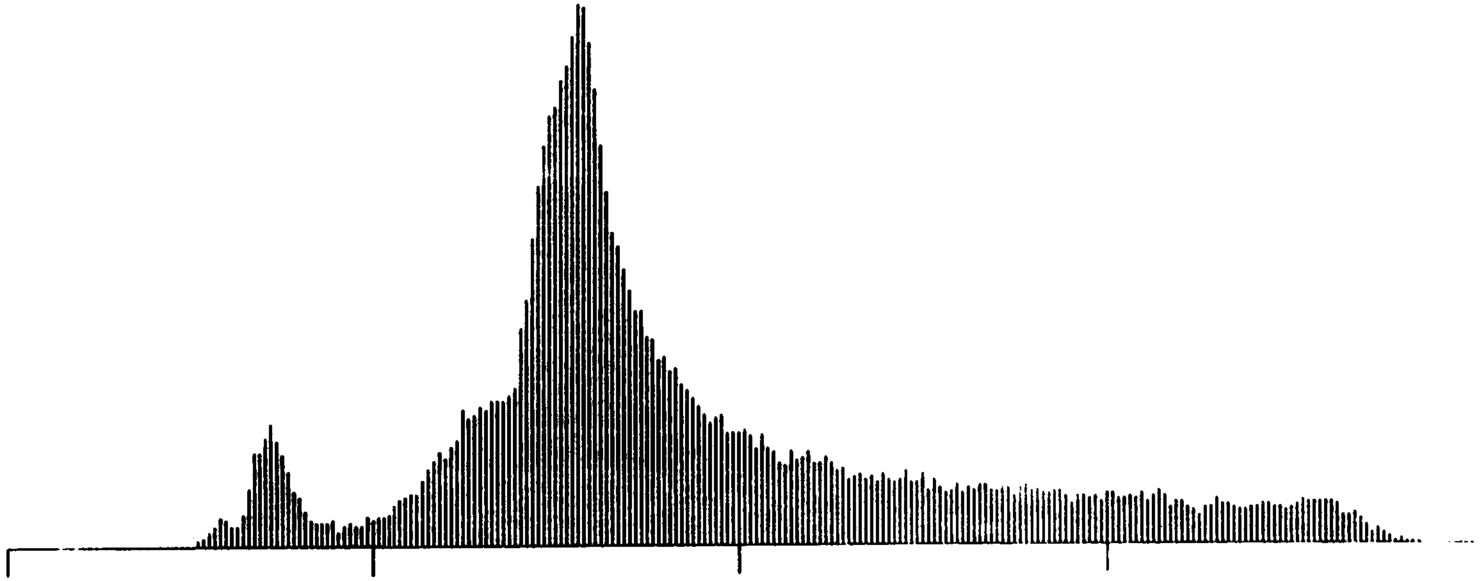
COMPANDER



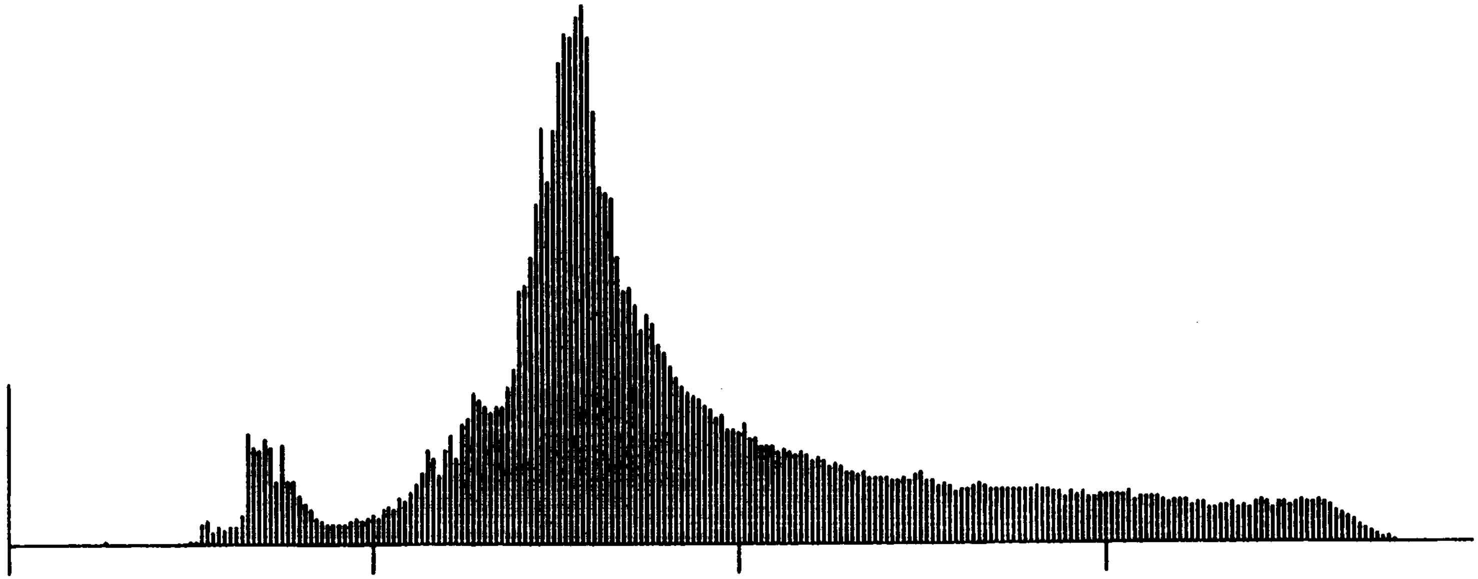
BLACK SATELLITE: ORIGINAL IMAGE



BLACK SATELLITE: COMPANDED AND PIXEL PAIRED

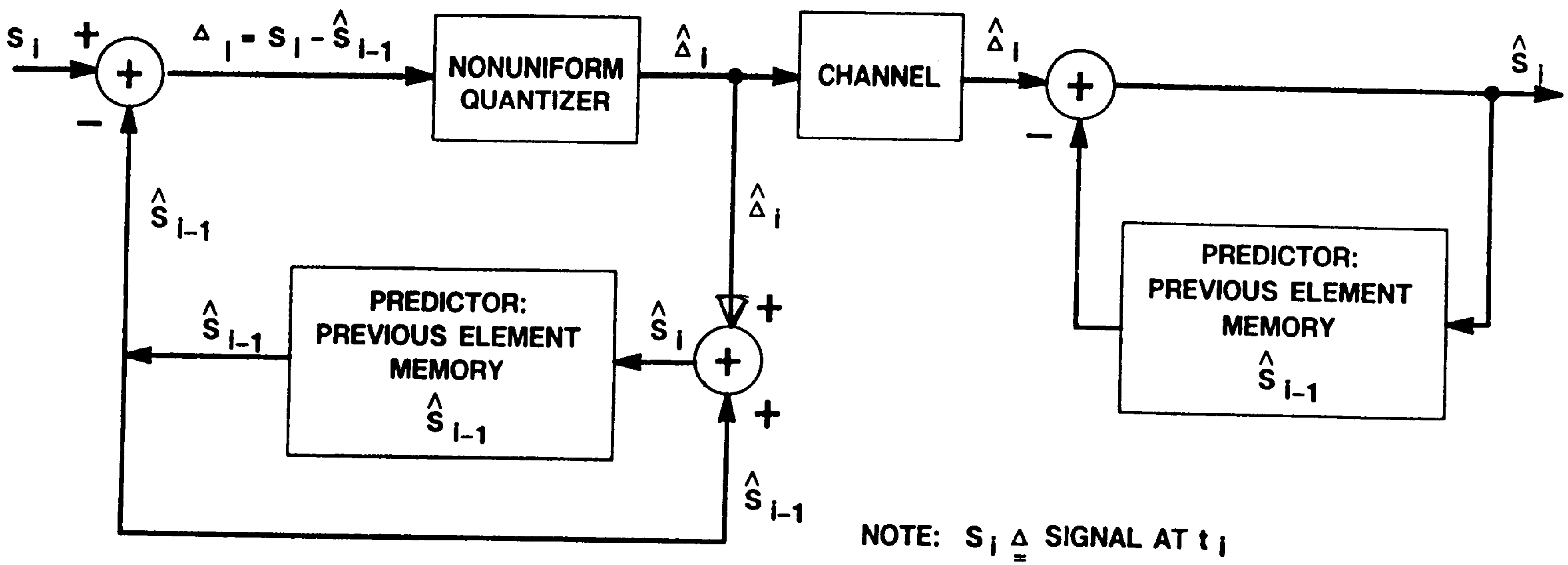


BLACK SATELLITE: ORIGINAL IMAGE



BLACK SATELLITE: COMPLETE ALGORITHM

BASIC DPCM ENCODER / DECODER



NOTE: $S_i \triangleq$ SIGNAL AT t_i

DPCM OPTIMIZATION – GLOBAL STATISTICS

* MEASURE CORRELATION COEFFICIENTS OF TYPICAL IMAGES

* OPTIMIZE PREDICTORS – MIN. LEAST SQUARE ERROR

* OPTIMIZE NON – UNIFORM QUANTIZER

* DESIGN HUFFMAN ENTROPY CODE

MEASUREMENTS CORRELATION COEFFICIENTS

−1,1	0,1	1,1
−1,0	0,0	1,0
−1,−1	0,−1	1,−1

−1,1	0,1	1,1
−1,0	X	

−1/2	+3/4	
+3/4	X	

POSSIBLE
PREDICTORS

0	+1/2	
+1/2	X	

OMV – DPCM PREDICTOR MODES

B	C
A	X

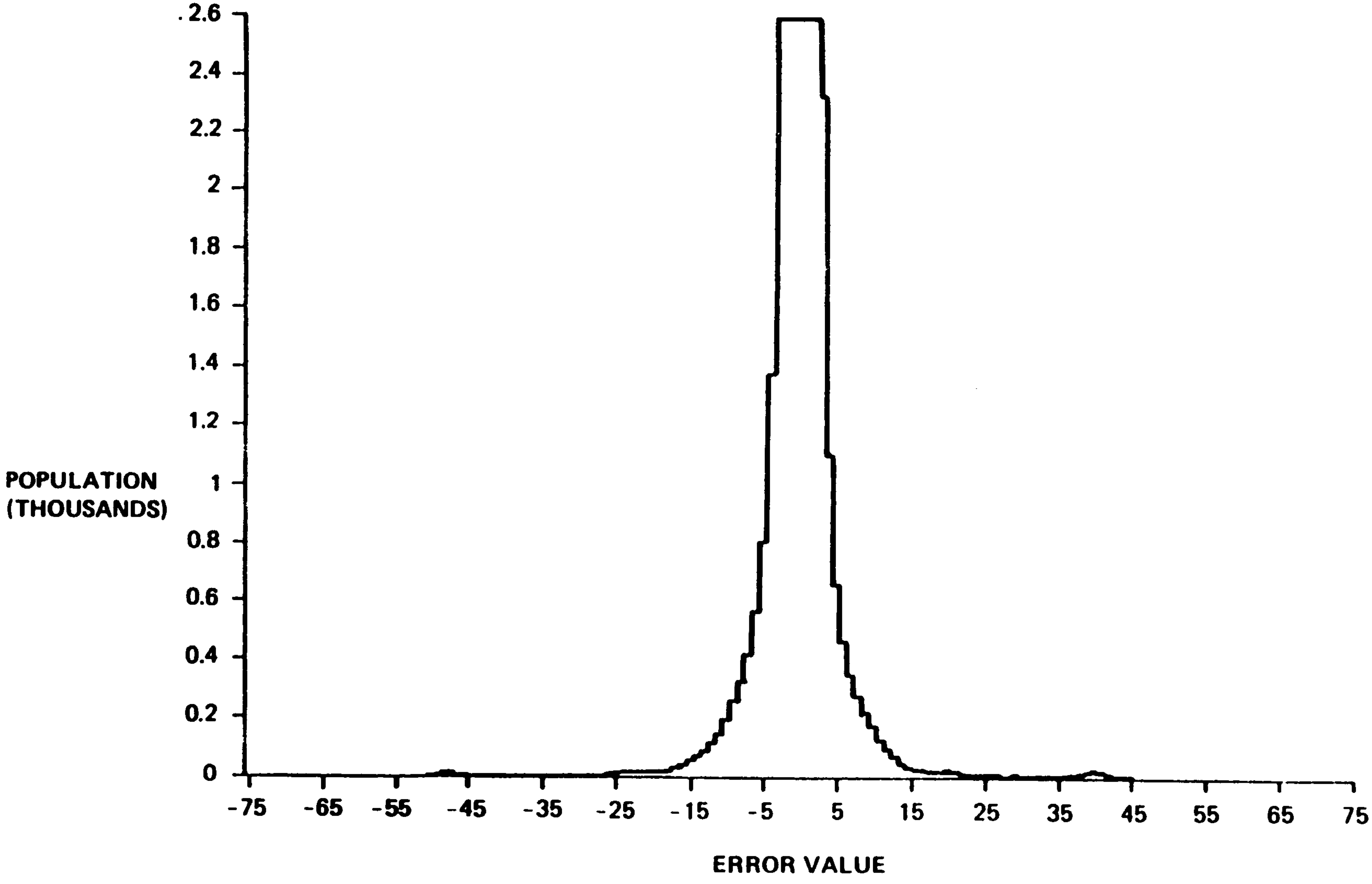
2–D NORMAL OPERATION ADAPTIVE

1–D NORMAL HORIZONTAL REFRESH 1 x A

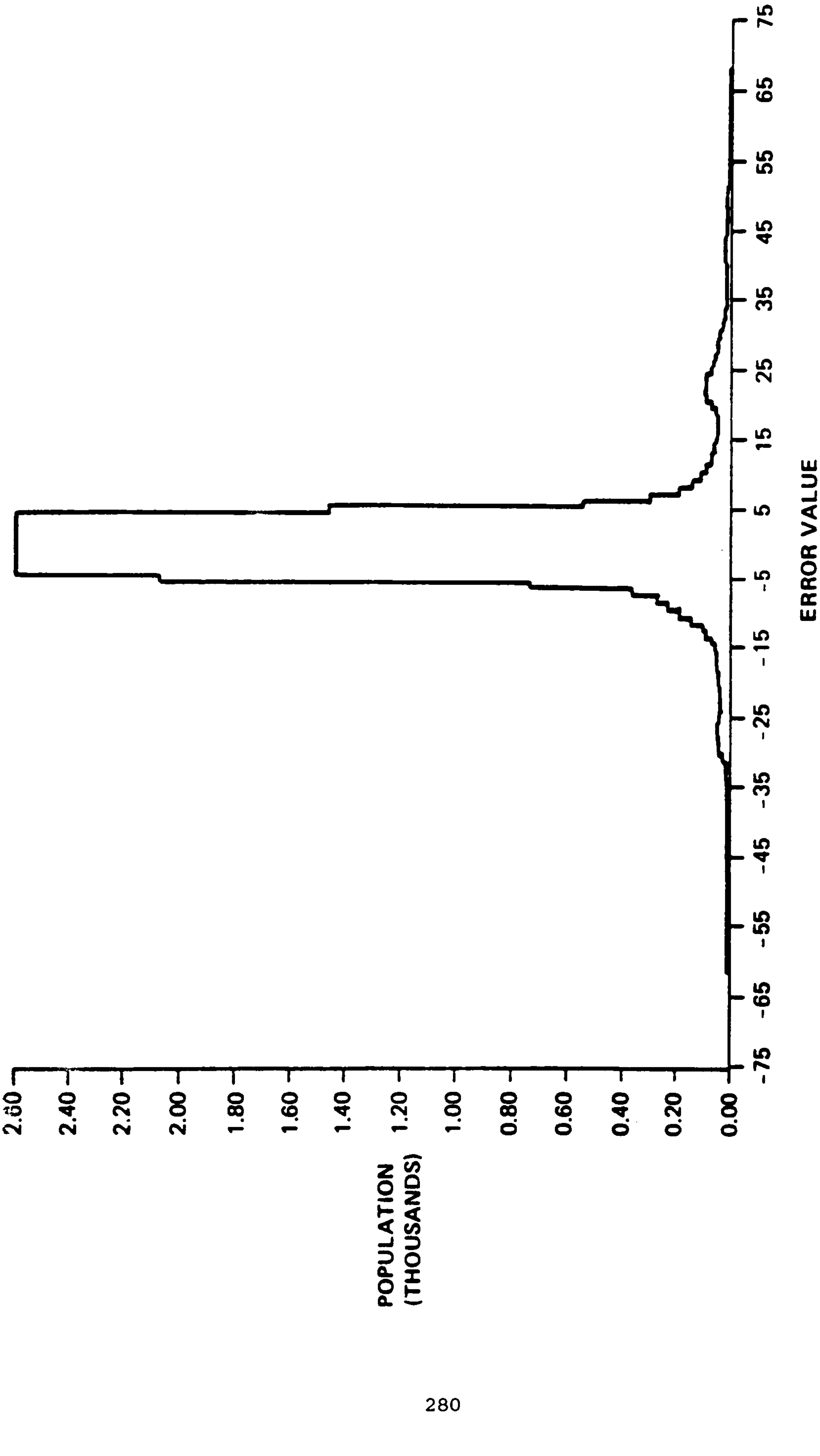
1–D START OF REMAINING LINES SUBFRAME 1 x C

ADAPTIVE DPCM BASED ON GLOBAL AND LOCAL CORRELATION

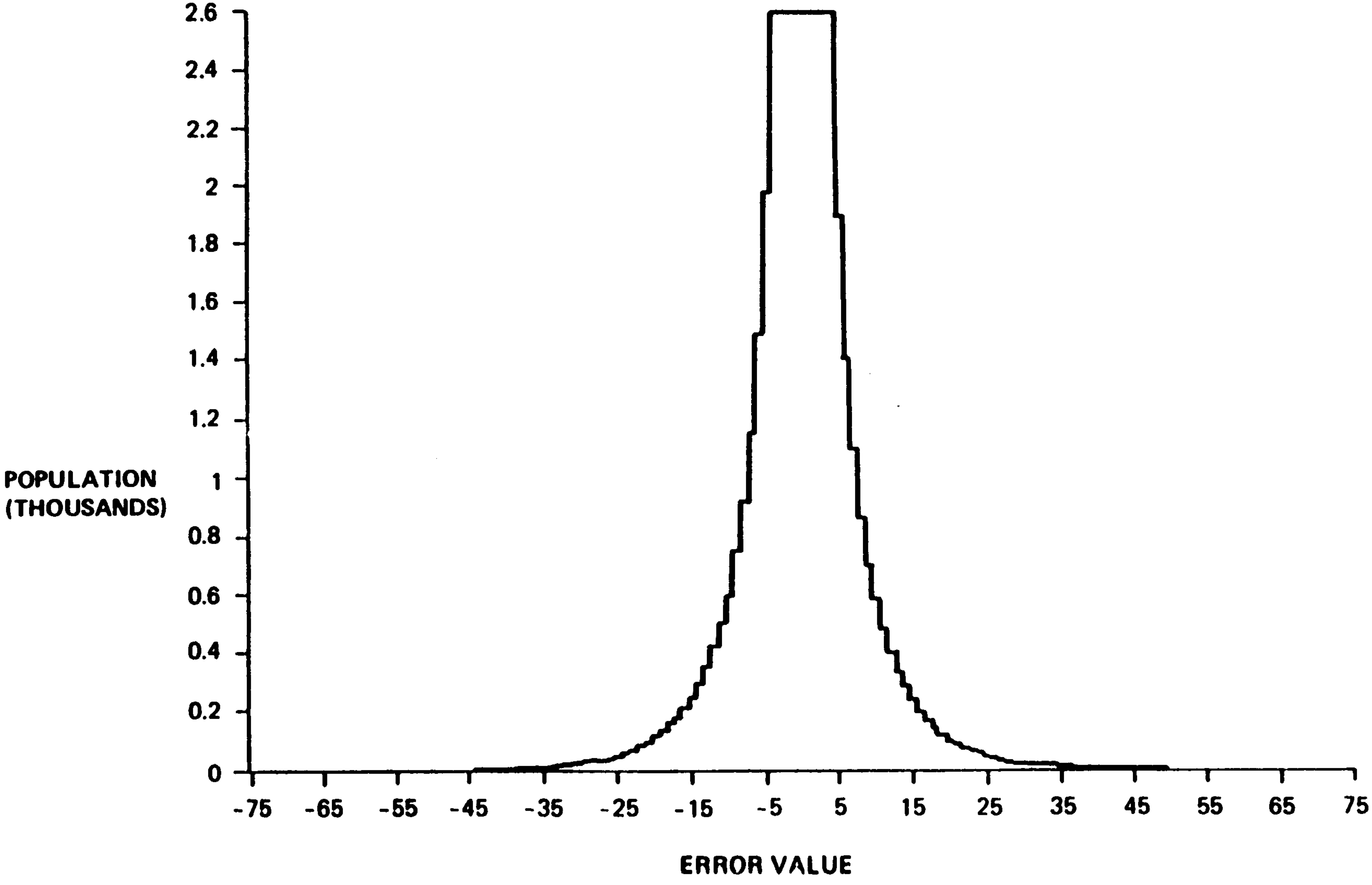
- TEST LOCAL CORRELATION
- $\geq$ THRESHOLD LOCAL PREDICTOR
- ALL OTHER CASES GLOBAL PREDICTOR



DOCKING TARGET WITH PIXEL PAIRING

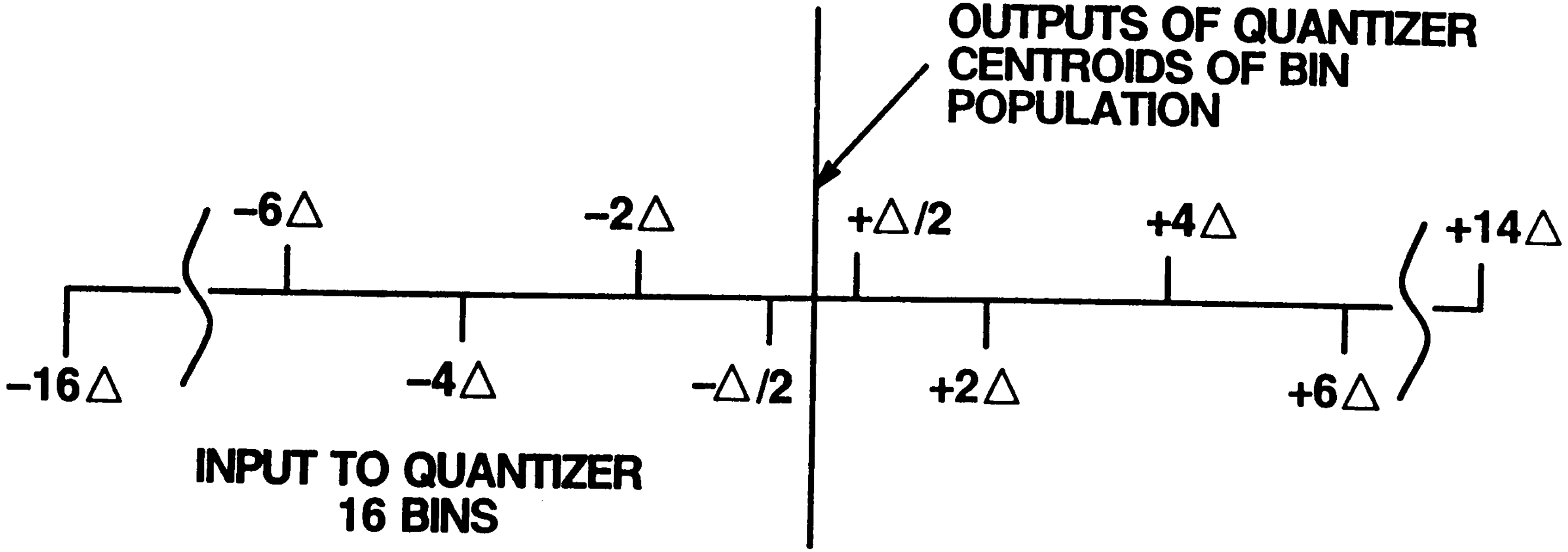


SPIN-SAT WITH PIXEL PAIRING



BLACK-SAT WITH PIXEL PAIRING

NON – UNIFORM QUANTIZER



OUTPUTS OF QUANTIZER
CENTROIDS OF BIN
POPULATION

INPUT TO QUANTIZER
16 BINS

NON – UNIFORM QUANTIZER OPTIMIZATION

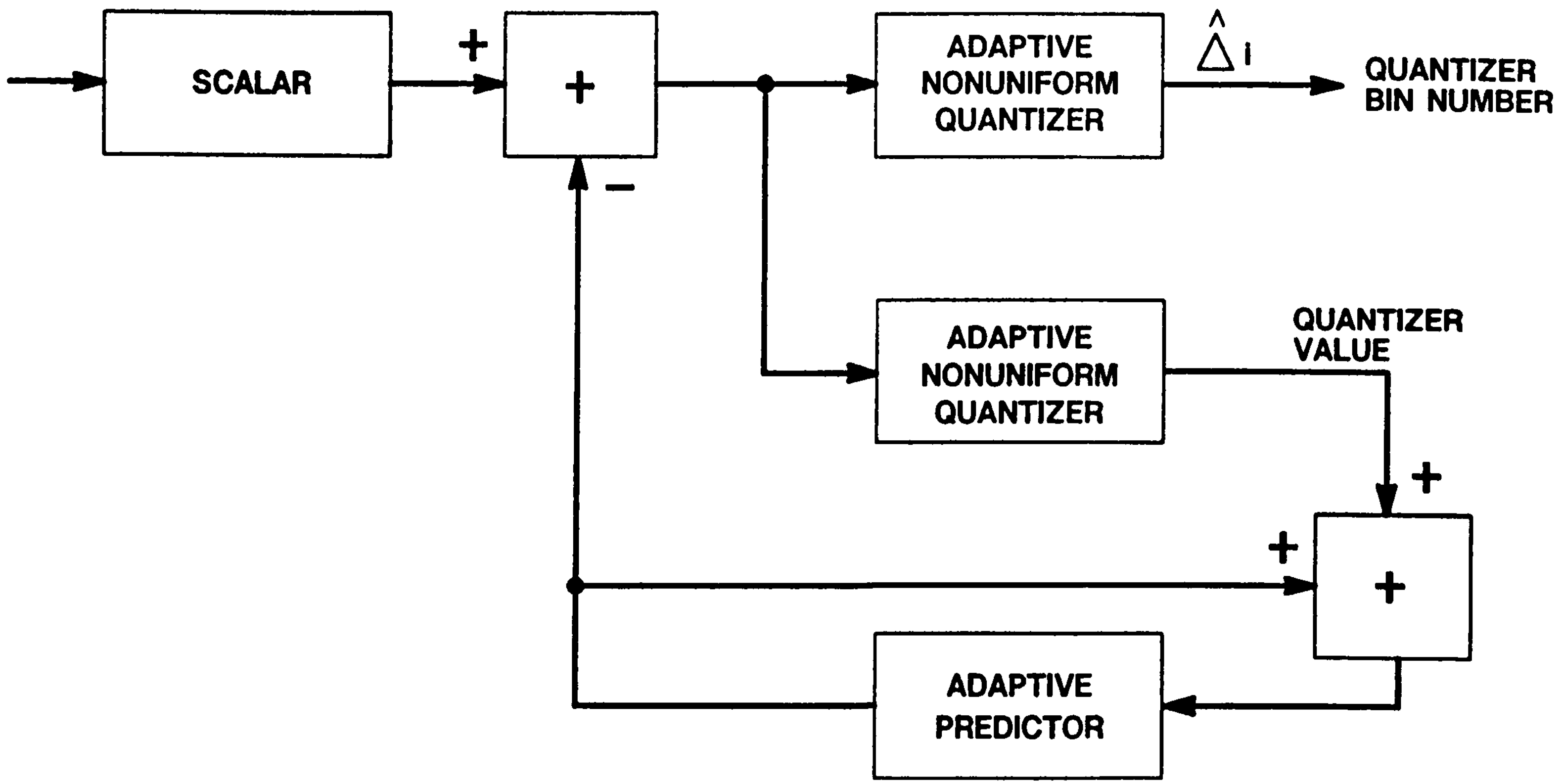
* SPACE IMAGERY ANALYZED – SCALAR VALUE VERSUS 1.3 BITS PER PIXEL

* STATISTICS TAKEN ON ERROR SIGNAL INTO NON–UNIFORM QUANTIZER

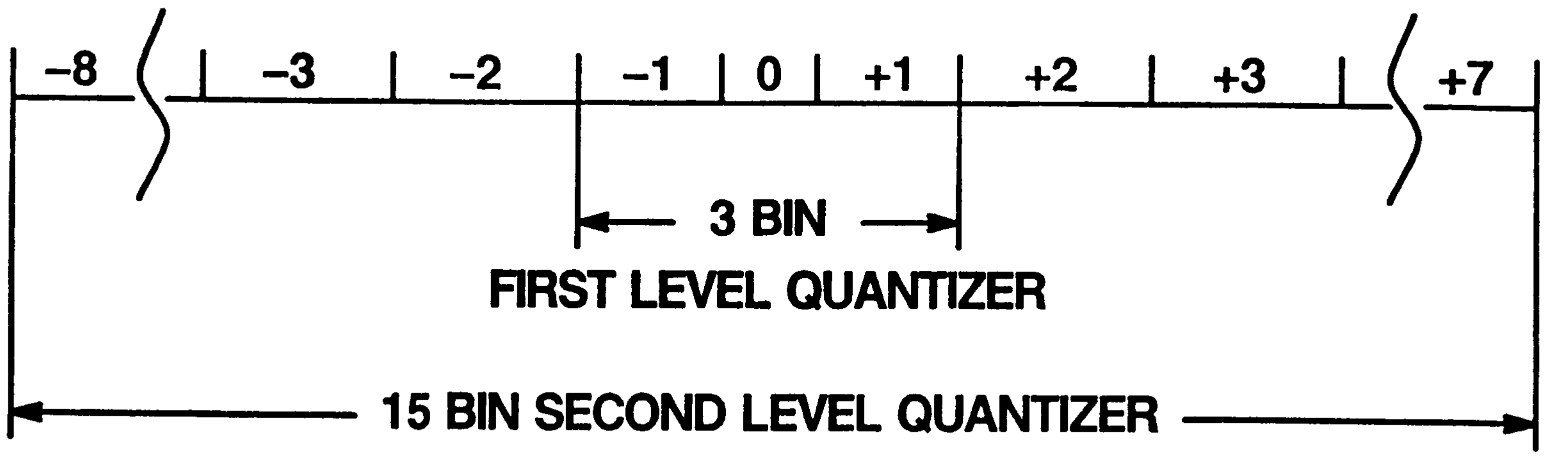
* ENSEMBLE AVERAGE OF MANY SCENES

* 16 NON–UNIFORM QUANTIZER TABLES DEVELOPED BY COMPUTING THE CENTROIDS OF QUANTIZER BINS

OMV – DPCM LOOP



DUAL – TIER QUANTIZATION ERROR SIGNAL



3 BIN
FIRST LEVEL QUANTIZER

15 BIN SECOND LEVEL QUANTIZER

VECTOR QUANTIZATION

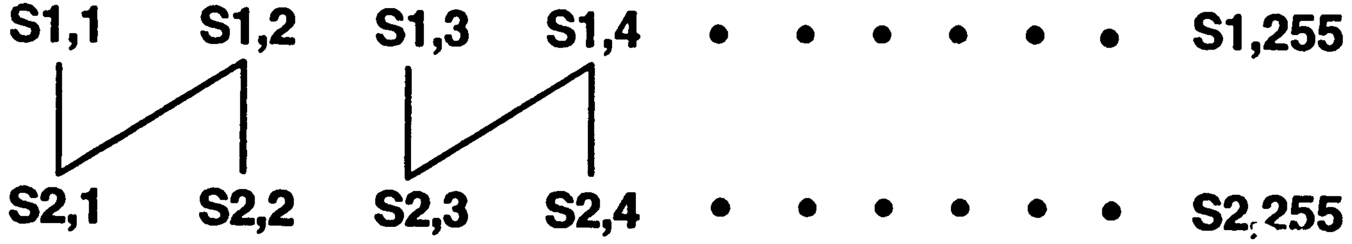
* TEST 2x2 ARRAY OF SCALARS LIE WITHIN –1, 0, +1

* QUANTIZER BINS REPRESENT 3 VALUES

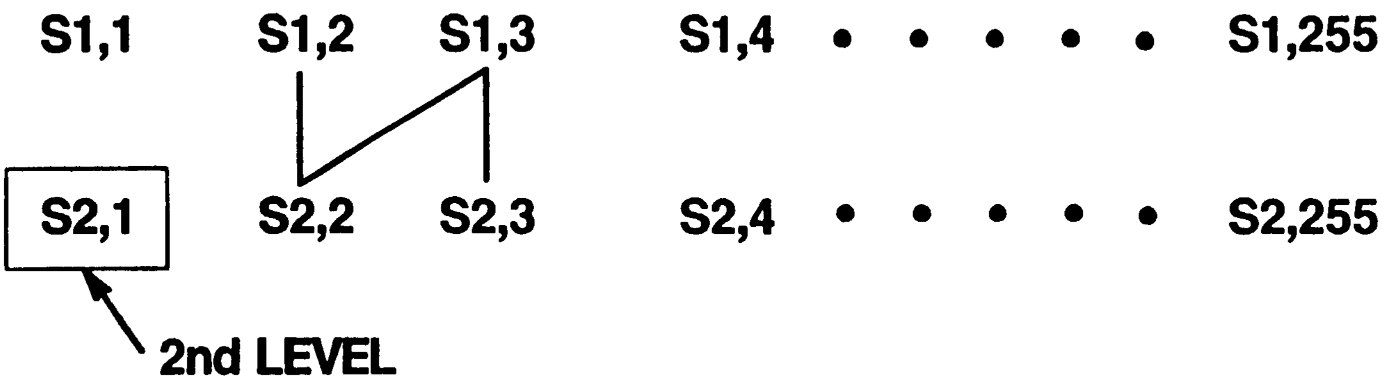
* 4 – SCALARS REPRESENT 4 DIMENSIONS

* THUS 3^4 OR 81 VECTORS MAPPED INTO M–1 CODE BOOK

VECTOR QUANTIZATION FOUR SCALARS WITHIN INNER QUANTIZER



INDIVIDUAL QUANTIZATION WHEN SCALARS HIT INTO SECOND LEVEL QUANTIZER



S1,1 S1,2 S1,3 S1,4 • • • • • S1,255

S2,1 S2,2 S2,3 S2,4 • • • • • S2,255

2nd LEVEL

*CODE S1,1	SCALAR	CODE BOOK M–2
*CODE S2,1	SCALAR	CODE BOOK M–2
*S1,2; S2,2;	S1,3; S2,3	VECTOR CODE BOOK M–1

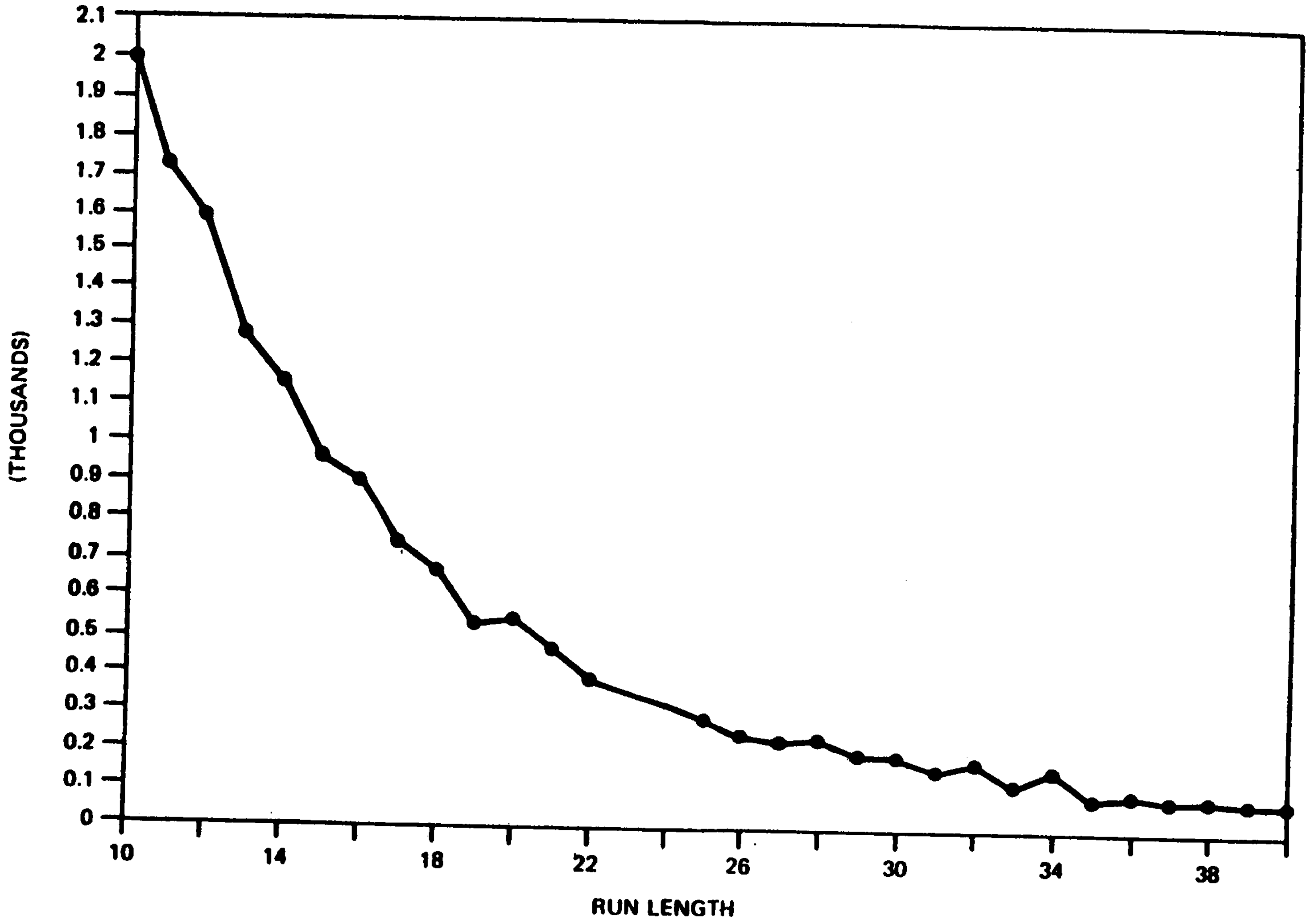
RUN LENGTH CODING

* RUN ≥ 10 BIN NUMBERS = 0

* RUN LENGTH ENCODE CODE BOOK M–3

* RUNS >74 CODE – 64

* START RUN LENGTH COUNTER AT 10


2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
(THOUSANDS)
10
14
18
22
26
30
34
38
RUN LENGTH

SCALAR = 8

ENTROPY ENCODER

* ENSEMBLE AVERAGING OF MANY SPACE SCENES

* CODE BOOKS M–1, M–2, M–3

* FREQUENCY OF OCCURRENCE USED TO DERIVE ONE HUFFMAN TABLE

* 4 – HUFFMAN TABLES ADAPTIVELY SWITCHED BASED ON SCALAR VALUES

RATE CONTROL ALGORITHM

* ELASTIC BUFFER MIDPOINT NORMALIZED TO ONE

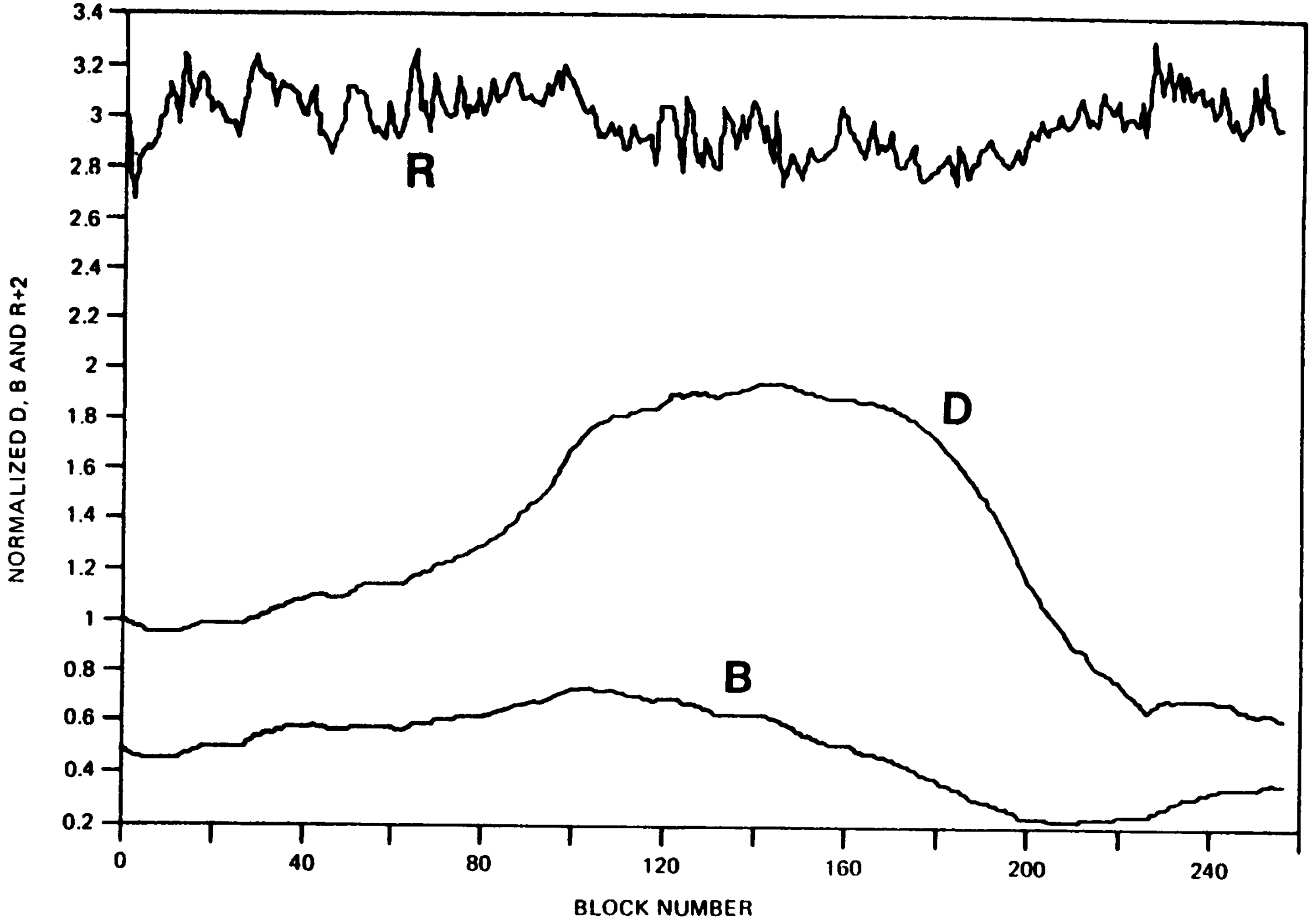
* EMPTY – 0.5   FULL + 0.5

* SCALAR CHANGED EVERY N – LINES

* MODULATE BUFFER OCCUPANCY ABOUT ONE

* SCALAR INCREASED REDUCES BITS / PIXEL

* UNDER FLOW AVOIDED BY BIT STUFFING WITH
  64 BIT UNIQUE WORD



GENERIC RATE CONTROL ALGORITHM

D, B AND R HISTORY

c-4

CHANNEL REQUIREMENTS FOR 10^{-5} BIT ERROR RATE OF 1/2 RATE VITERBI DECODER

- CARRIER TRACKING LOOP 12.4 dB
- COSTAS LOOP – PHASE ERROR 1.9 dB – JITTER .3 dB
- E_b / N_o INTO DEMODULATOR 6.8 dB
- VITERBI INPUT 4.6 dB

RS – 254,238 CODE EXTENDED TO 255,238 WITH SYNC

	SYMBOLS	BITS/SYMBOL	TOTAL BITS
MESSAGE	238	X 8 =	1,904
PARITY	16	X 8 =	128
SYNC	1	X 8 =	8
	ONE RS CODE WORD= 2,040 BITS		

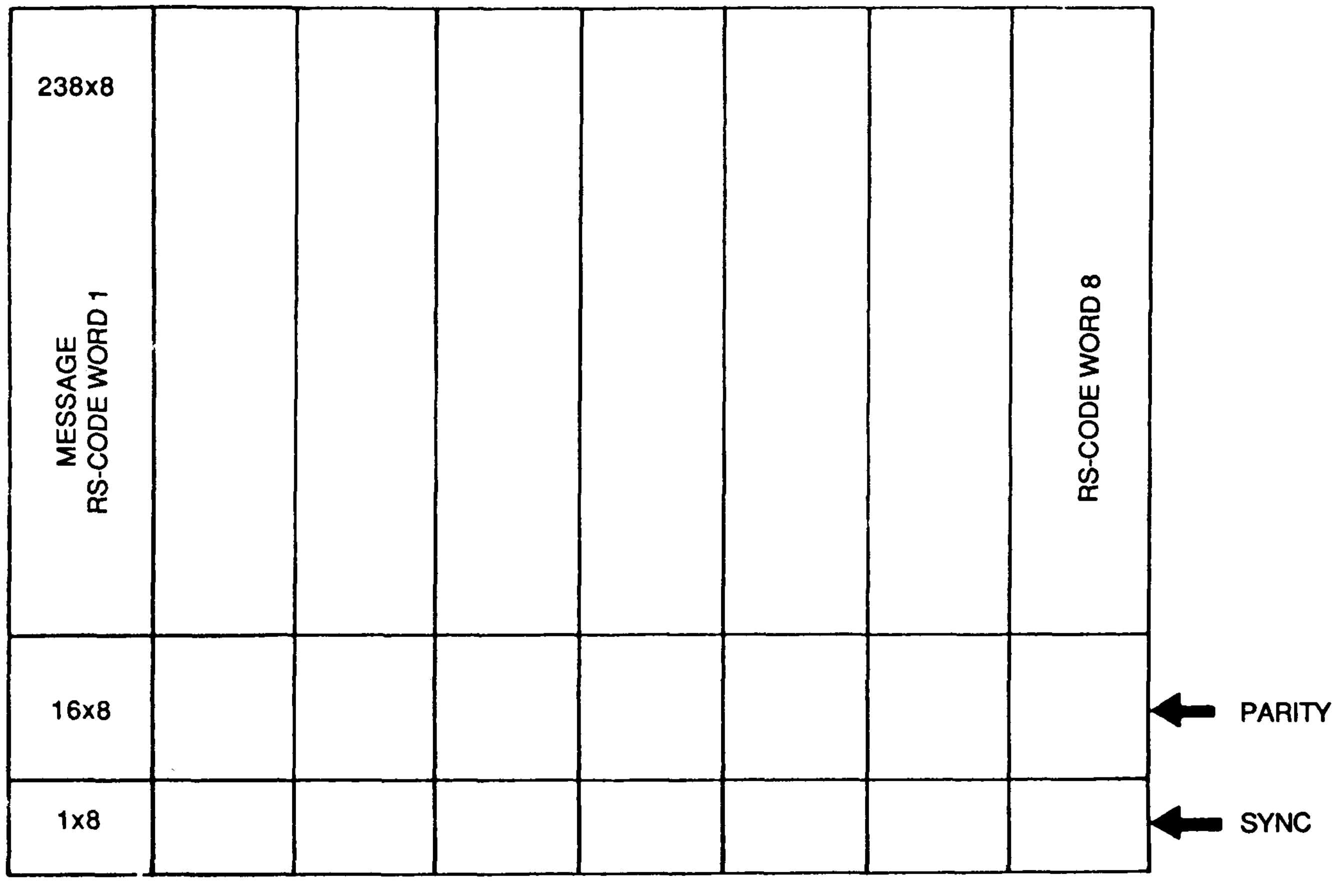
INTERLEAVED TO A DEPTH OF 8

2,040 X 8 = 16,320 BITS IN ONE CODE BLOCK

CHANNEL STATISTICS

* STUDY BY LIN   COM FOR GODDARD

* INTERLEAVING TO DEPTH 8 SUFFICIENT

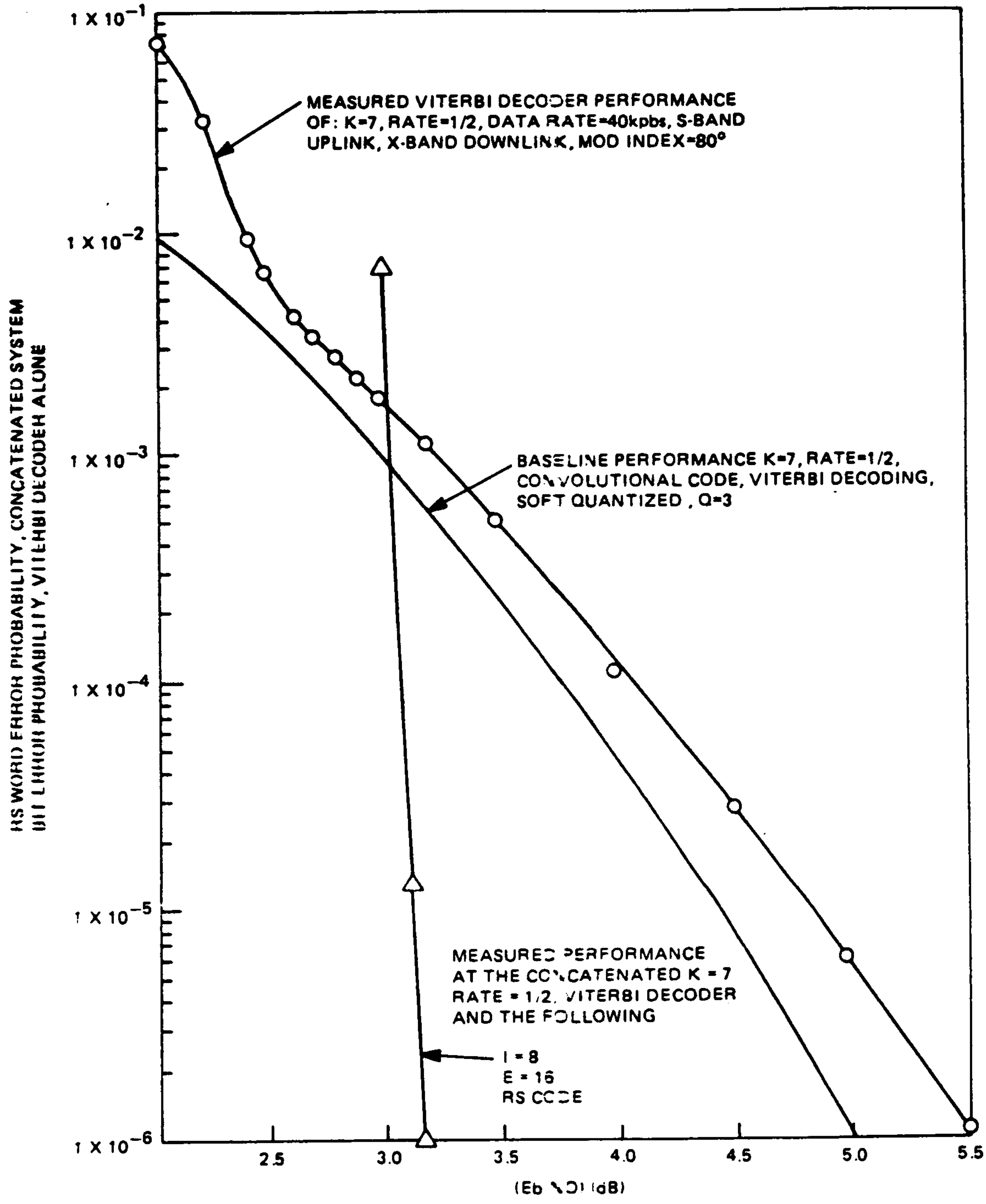


RS–255,238 CODE BLOCK,
STANDARD BLOCK INTERLEAVE

- IF THE ACTUAL NUMBER OF ERRORS, s, AND THE ACTUAL NUMBER OF ERASURES, t, IN THE CODE WORD IS SUCH THAT t, $\leq$ 16, BUT 2s + t > 16 THERE CAN BE A MISDETECTION; THE CONDITIONAL PROBABILITY OF THIS MISDETECTION IS;

$$P\left[\, t \leq 16 \text{ , BUT } 2S + t > 16 \Big/ \frac{1}{\left[\frac{(16-t)}{2}\right]!} \,\right]$$

MISDETECTIONS



PERFORMANCE OF THE RS / VITERBI CONCATENATED CODING SYSTEM (WORD ERROR PROBABILITY VERSES Eb/No)

ERROR CONTAINMENT

- * COUNT NUMBER OF DECODED HUFFMAN CODE WORD IN SUB – FRAME
- * COUNT CORRECT – USE NEW SUB – FRAME DATA
- * COUNT INCORRECT – REPEAT OLD SUB – FRAME
- * SUB – FRAME SIZE ADJUSTABLE BY GCC
- * BIT ERROR MONITOR FOR OPERATOR

N89-22343

LOCAL INTENSITY ADAPTIVE IMAGE CODING

Friedrich O. Huck
NASA Langley Research Center

The objective of preprocessing for machine vision (low-level vision processing) is to extract intrinsic target properties. The most important properties ordinarily are structure (contour outlines or primal sketches) and reflectance (color). Illumination in space, however, is a significant problem as the extreme range of light intensity, stretching from deep shadow to highly reflective surfaces in direct sunlight, impairs the effectiveness of standard approaches to machine vision. To overcome this critical constraint, we are investigating an image coding scheme which combines local intensity adaptivity, image enhancement, and data compression. It is very effective under the highly variant illumination that can exist within a single frame or field of view, and it is very robust to noise at low illuminations.

In this presentation, I will 1) review some of the theory and salient features of the coding scheme, 2) characterize its performance in a simulated space application, and 3) describe our research and development activities.

The local intensity adaptive image coding consists of an innovative model of processing in the human retina referred to as Intensity Dependent Spread (IDS)(1) and some additional logic to extract contour outlines and reflectance ratios at the boundary of two surfaces. Figure 1 is a schematic representation of the IDS model. The line of detectors represents a slice through a two-dimensional array of detectors. When an optical image, or light distribution, falls on the detector array, then each detector sends a signal into a network, where it spreads out. Each channel, in turn, sends out a signal that is the sum of all the signals that arrive in its location in the summation network. The special property of the IDS model has to do with the way that the signal from each detector spreads in the summation network. As depicted in the lower half of Figure 1, the magnitude of the signal at its center is proportional to the intensity of the light falling on the detector, and the spread of the signal is inversely proportional to this intensity. The total volume under the spread remains constant. That is all there is to the model. It has been demonstrated that this simple space-variant model of image processing has many of the properties of human visual perception(2).

Figure 2 demonstrates the response of the IDS processor to a spot, or point source, that is brighter than the background. Each detector spreads its signal as governed by the intensity of the light falling on the detector. All of the spreads for the uniform background are the same except for the one detector that is more brightly lighted. Its spread is higher and narrower. Each output channel just adds up all of the contributions it receives. The result of this processing is shown as the output signal. As can be seen, the IDS response to a point source has the same shape as the response of Marr and Hildreth's(3) familiar Laplacian of a Gaussian (∇^2G) operator for enhancing edges. In fact, the IDS processor exhibits center surround antagonism and all other manifestations of bandpass filtering that have made the ∇^2G operator a favorite algorithm for low-level vision processing. However, the IDS response is nonnegative and spatially variant. As we will show in the next three figures, the IDS processor accounts for several familiar perceptual phenomena of human vision that make it a highly robust low-level vision operator.

First, let us compare the IDS operation to conventional imaging. Figure 3 shows intensity profiles taken across conventional and IDS images of a step-type edge input for three illuminations, or SNR's. Conventional image-gathering yields a blurred representation which is visually representative of the target if the SNR is sufficiently high. As the illumination decreases, the representation gets buried in the noise. Image gathering with the IDS processor yields a target representation that consists of pulses. The one-crossing of each pulse locates the position of an edge in the target. The peak and trough values of the pulse are proportional to the ratio of the reflectances at the two sides of the edge, entirely independent of illumination. As the illumination decreases, the width of the pulse becomes broader (thereby trading resolution for sensitivity), but the accuracy of the one-crossing is unimpaired. For machine vision, this property means that edge detection for determining structure is highly robust to widely variant illumination.

Next, let us compare the IDS operation to edge detection with the linear Laplacian of a Gaussian (∇^2G) operator as well as conventional imaging. Figure 4 shows intensity profiles taken across conventional images and outputs from the ∇^2G and IDS operators for two illuminations, high and low. Noise is disregarded for simplicity. The peak and trough values of the ∇^2G pulses are proportional to both illumination and reflectance. It is therefore not possible to characterize the reflectance properties of the target independent of illumination. However, the peak and trough values of the IDS pulses are proportional only to the reflectance changes. This striking property of the IDS processor mimics human visual perception (Weber's law). For machine vision, this property means that it is possible to extract the reflectance ratio at the boundary of two areas.

And finally, let us turn to a scene which realistically simulates a scene that may be encountered in space. Figure 5 compares the IDS operation to conventional image gathering and edge detection with the ∇^2G operator. A single source of light (35mm slide projector) was used to illuminate the model of a satellite and astronaut surrounded by a black curtain. The image was obtained with a standard 640-by-484 sensor-array camera using 8-bit encoding. By being locally adaptive to both the directly illuminated part of the satellite as well as to the astronaut located in deep shadow, the IDS operator provides a much more complete and reliable rendition of the scene.

If we use the $\nabla^2 G$ processor and zero-crossing detection logic, then we need to transmit only the location of the (correctly and incorrectly) identified edge locations. The corresponding data compression is 24. If we use the IDS processor together with one-crossing detection logic, then the data compression ratio is 20, similar as before. However, in addition to this data compression, the astronaut, who was nearly lost before, is now clearly detected and could probably be identified by a higher level AI algorithm. It should also be noted that we did not yet attempt to use efficient coding techniques for transmitting contour outlines. To do so would probably lead to further significant increases in compression.

The identification of the scene from structure alone could be significantly improved upon when information about reflectance ratios is retained in the transmitted data. The data compression would then reduce from 20 to 13. Figure 6 presents three representations that can be extracted from the complete IDS data. As before, we could display structure alone, or we could display reflectance changes as well as structure. We could draw the contour outlines on the restored gray levels in white to emphasize structure or in black to enhance visual sharpness. In either case, it is important to note that we have not sacrificed our accuracy of edge location by the data compression, and that we have been able to extract reflectance changes independent of illumination. In fact, it is possible under suitable conditions to locate edges with higher accuracy than the sampling intervals of the camera. The only sacrifice we have made is to trade discrimination of fine detail for increased sensitivity to adapt to low illuminations. However, this fine detail would otherwise often be lost in noise. An important extension of IDS processing would be to extract color. Color could then be correctly detected independent of illumination.

We must now admit that the structure-plus-reflectance images in Figure 6 are fakes to illustrate a potential capability. They simply represent the superposition of (correctly extracted) structure on regular (brightness) images. We are now extending our structure extraction algorithm to an improved structure-plus-reflectance extraction algorithm.

The IDS model of retinal processing was conceived by Tom Cornsweet of the University of California, Irvine. Langley Research Center (LaRC) is now working with Cornsweet and Odetics, Inc., in the evaluation of this model for various applications of interest to NASA, including, in particular, machine vision and image coding for space operations. Odetics is also under contract to LaRC to develop a hardware implementation of the IDS processor (see Figure 7). This processor will be capable of handling image data at real-time TV rates (30 frames per second). It will be implemented on several boards for the DATACUBE of Sun image-processing work stations. This board will become commercially available in the Fall 1989.

The full potential of the IDS processor for data compression as well as image enhancement and feature extraction is realized, of course, only when it is implemented as a focal-plane processor, or "retinal camera" (see Figure 8). The present design of the IDS processor for Sun workstations could be implemented in one 5" by 5" board with 8 VLSI chips. A more advanced approach would be parallel asynchronous focal-plane image processing (see Figure 9). This processor is representative of a new class of devices which would permit full two-dimensional parallel readout and processing perpendicular to the focal plane. Advantages over conventional image gathering and processing techniques include rapid parallel distributed processing, high dynamic range, and the elimination of conventional charge transfer, multiplexing, and preamplifiers. Vision processing could potentially be performed several orders of magnitudes faster than with conventional approaches. Moreover, parallel processing would be ideal for tasks like visual pattern recognition. However, the development of this approach is still in its initial experimental stage.

REFERENCES

1) Cornsweet, T.N. and J. I. Yellott, Jr., "Intensity-dependent spatial summation," J. Opt. Soc. Am. 2, pp. 1769-86 (1985).

2) Cornsweet, T.N., Visual Perception, Academic Press, New York, 1970.

3) Marr and Hildreth, "Theory of edge detection," Proc. R. Soc. London B 207, pp. 187-217 (1988).

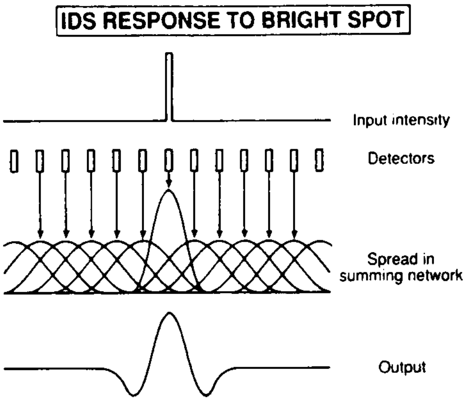


Figure 1

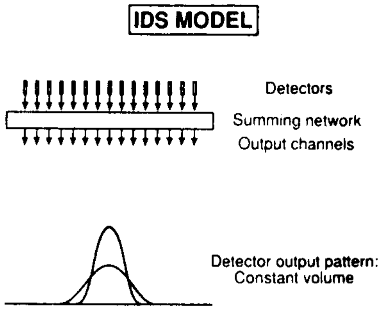


Figure 2

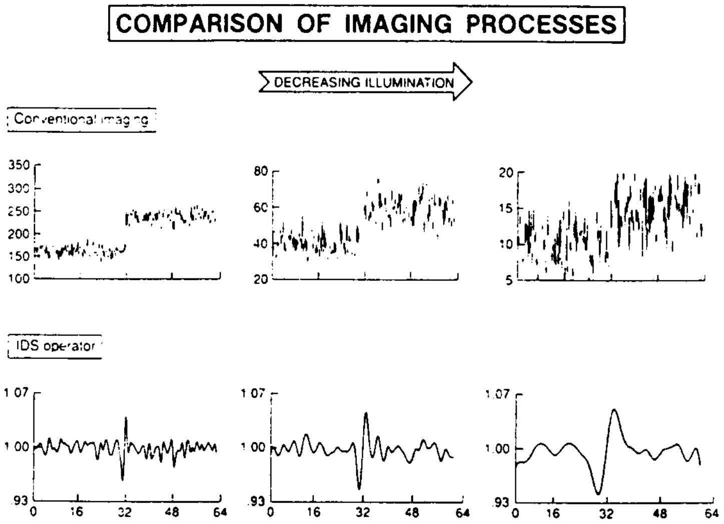


Figure 3

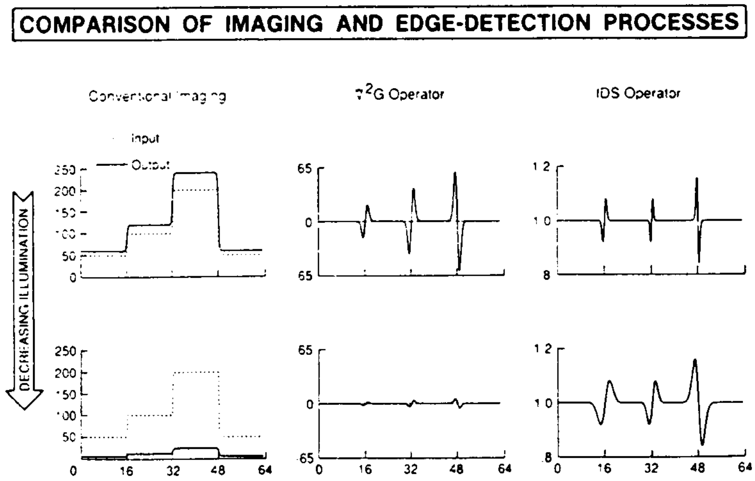
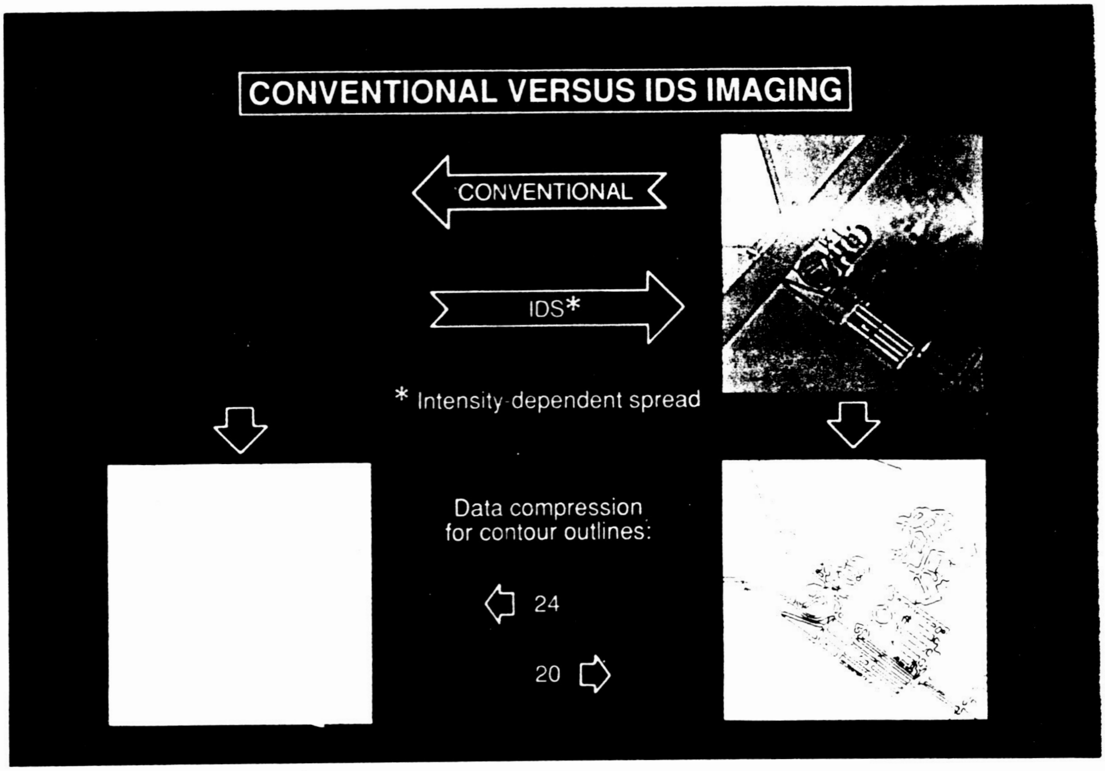


Figure 4



CONVENTIONAL VERSUS IDS IMAGING

CONVENTIONAL

IDS*

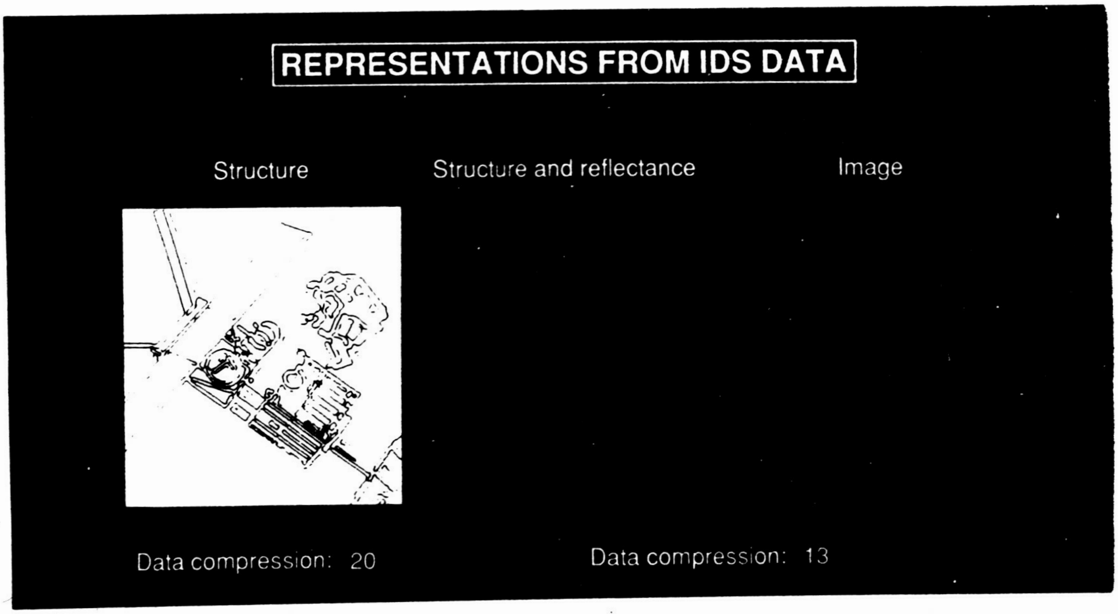
* Intensity-dependent spread

Data compression
for contour outlines:

24

20

Figure 5



REPRESENTATIONS FROM IDS DATA

Structure

Structure and reflectance

Image

Data compression: 20

Data compression: 13

Figure 6

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

REALTIME IDS VISION WORKSTATION

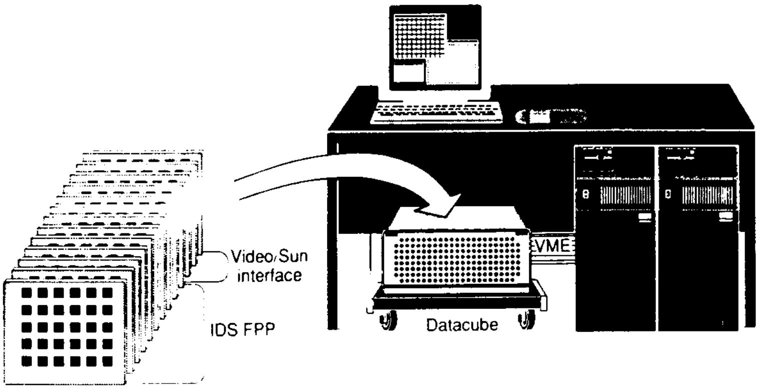


Figure 7

IDS FOCAL-PLANE PROCESSING CAMERA
(VLSI)

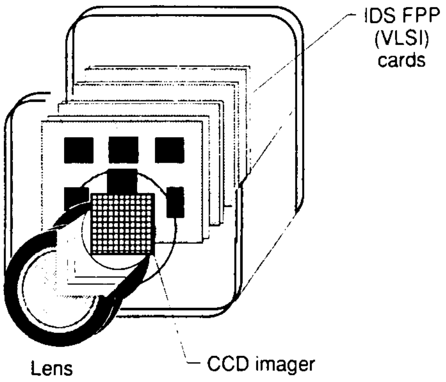
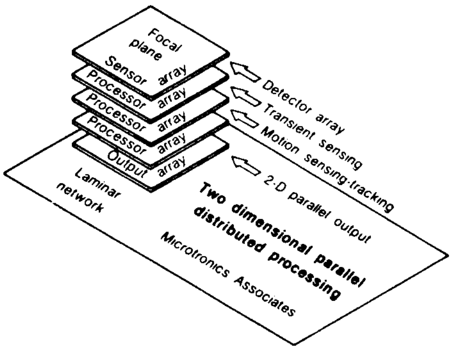
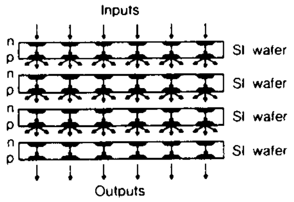


Figure 8

PARALLEL ASYNCHRONOUS FOCAL-PLANE IMAGE PROCESSOR



- 2-D parallel distributed data flow perpendicular to focal plane



- Low power, low noise integrate-and-fire devices for low level current-to-frequency conversion in every processing channel and at every network mode

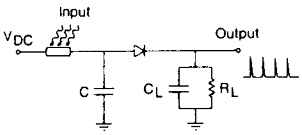


Figure 9

DATA COMPRESSION EXPERIMENTS WITH LANDSAT THEMATIC MAPPER AND NIMBUS-7 COASTAL ZONE COLOR SCANNER DATA

James C. Tilton and H. K. Ramapriyan
Space Data and Computing Division
Goddard Space Flight Center

ABSTRACT

The variety of remote sensing instruments expected to be deployed in the last decade of this century and early 21st century, their resolutions and the anticipated data collection rates imply requirements for on-board reduction of data volumes in order to maximize the scientific return from space in the face of limited transmission bandwidth. Such data reductions can be achieved either through lossless or lossy data compression or through on-board "analysis and information extraction" and transmission of results. Several recent and potentially anticipated advances in computer science and hardware technology make it feasible to consider the development of on-board computer systems with sufficient capability to accomplish the above tasks. It is obvious that compression techniques which are shown to be feasible for on-board implementation can also be implemented for on-ground data compression thus helping reduce the archival storage costs, increase the on-line availability of data, and reduce times needed for browsing data for a given region or time interval.

Studies evaluating image entropies treating images pixel by pixel or considering differences between adjacent pixels indicate that lossless compression ratios of 1.5 to 3 can be achieved (of course, depending on the data) by optimal encoding of pixel values or differences. It is to be noted, however, that the entropies so defined do not represent the theoretical performance limit on reversible (i.e., lossless) data compression. Lossy compression

techniques such as predictive encoding, discrete transforms, cluster coding and vector quantization can achieve greater compression ratios which are a function of the acceptable level of loss. A common objection to these data compression techniques is that with significant compression (factors greater than 10) the data cannot be exactly recovered in their raw form.

For any lossy technique to be acceptable for a given application, it is necessary to demonstrate that the most of the relevant information remains in the compressed data. Therefore, to prove the utility of a compression technique for a scientific application, it is necessary to perform case studies with remotely sensed data in selected disciplines, use well accepted analysis techniques, and demonstrate that compressed data result in very nearly the same analysis results as the original data. With sufficient interaction between the scientific community and developers of data compression techniques it should be possible to define such case studies, and in fact, arrive at techniques which will not only reduce the data volume using criteria tailored to the analysis techniques, but also facilitate data analysis by direct use of compressed data.

In this paper, we present a case study where an image segmentation based compression technique is applied to Landsat Thematic Mapper (TM) and Nimbus-7 Coastal Zone Color Scanner (CZCS) data. The compression technique, called Spatially Constrained Clustering (SCC), can be regarded as an adaptive vector quantization approach. The SCC can be applied to either single or multiple spectral bands of image data. The segmented image resulting from SCC is encoded in small rectangular blocks, with the "codebook" varying from block to block. Lossless compression potential (LCP) of sample TM and CZCS images are evaluated. For the TM test image, the LCP is 2.79. For the CZCS test image the LCP is 1.89, even though when only a cloud-free section of the image is considered the LCP increases to 3.48. Examples of compressed images are shown at several compression ratios ranging from 4 to 15. In the case of TM data, the compressed data are classified using the Bayes' classifier. The results show an improvement in the similarity between the classification results and ground truth when compressed data (with compression ratios of up to 13.8) are used, thus showing that compression is, in fact, a useful first step in the analysis. Future work in this case study will include the use of SCC-compressed CZCS data to obtain chlorophyll concentrations using the algorithm currently in use at GSFC for the production of global chlorophyll maps.

INTRODUCTION

The resolutions and anticipated data collection rates of the variety of remote sensing instruments expected to be deployed in the last decade of this century and the early 21st century imply requirements for on-board reduction of data volumes in order to maximize the scientific return from space in the face of limited down-link transmission bandwidth. Such data reductions can be achieved either through lossless or lossy data compression or through on-board analysis and information extraction and transmission of results. Several recent and anticipated advances in computer science and hardware technology make it feasible to consider the development of on-board computer systems with sufficient capability to accomplish the above tasks. It is obvious that compression techniques which are shown to be feasible for on-board implementation can also be implemented for on-ground data compression for the purpose of reducing archival storage costs, increasing the online availability of data, and reducing the time needed for browsing data for a given region or time interval.

Studies evaluating image entropies treating images pixel by pixel or considering differences between adjacent pixels indicate that lossless compression ratios of 1.5 to 3.0 can be achieved (of course, depending on the data) by optimal encoding of pixel values or differences (Chen[1], Ramapriyan[7], Wharton[11]). The entropies so defined do not necessarily represent the theoretical performance limit on reversible (i.e., lossless) data compression. However, the actual theoretical performance limit is likely to be less than twice the compression ratios indicated. Lossy compression techniques such as predictive encoding, discrete transforms, cluster coding and vector quantization can achieve greater compression ratios subject to an acceptable level of loss. A common objection to these data compression techniques is that with significant compression (factors greater than 10) the data cannot be exactly recovered in their raw form.

For any lossy technique to be acceptable for a given application, it is necessary to demonstrate that most of the relevant information is retained in the compressed data. To prove the utility of a compression technique for a scientific application we must perform case studies with remotely sensed data in selected disciplines, use well accepted analysis techniques, and demonstrate that the use of compressed data produces very nearly the same analysis results as with the original data. There are several case studies where effects of data compression on multispectral classification have been studied (Kauth[5], Hilbert[2], Ramapriyan[7]). A common characteristic among these studies is that the compression technique is really a precursor to analysis and information extraction. A variety of such case studies are needed for several scientific disciplines and applications with sufficient interaction between the scientific community and developers of data compression techniques. Through such case studies it should be possible to arrive at techniques which will not only reduce the data volume using criteria tailored to the analysis techniques, but also facilitate data analysis by direct use of compressed data.

In this paper we present a case study where an image segmentation based compression technique is applied to Landsat Thematic Mapper (TM) and Nimbus-7 Coastal Zone Color Scanner (CZCS) data. When accompanied by an encoding of the resulting segmentation, the Spatially Constrained Clustering (SCC) segmentation approach can be regarded as an adaptive vector quantization approach to data compression. The SCC data compression approach can be applied to either single or multiple spectral bands of image data. The segmented image resulting from SCC is encoded in small rectangular blocks, with the "codebook" varying from block to block.

ALGORITHMS AND ERROR MEASURES

In this section, we define the Lossless Compression Potential (LCP) of an image and the error measures used to evaluate the compressed data. We also describe the SCC segmentation algorithm and the block cluster coding method used to encode the segmented image to obtain a compressed image.

Lossless Compression Potential

Ideally, the LCP of a given image would be defined as the maximum factor by which the image can be reversibly (that is, losslessly) compressed. If every pixel of an image is totally uncorrelated with other pixels in the image, this ideal LCP could be easily calculated from the zeroth order entropy of the image. However, image pixels are generally highly correlated, causing the zeroth order entropy to underestimate the LCP of the image. To compensate partially for the correlation between image pixels, we define our LCP based on the zeroth order entropy of a difference image in which each pixel is represented as a function of three neighboring pixels and itself (Rosenfeld and Kak[8]):

$$d'(x,y) = d(x,y) - d(x-1,y) - d(x,y-1) + d(x-1,y-1) \qquad (1)$$

where d(x,y) represents the original image value at pixel (x,y), and d'(x,y) represents the difference image value at pixel (x,y). This is a special case of two-dimensional Differential Pulse Code Modulation (DPCM) (Jain[3]). (Note: the first row and column of d'(x,y) are generated by assuming that the "0^{th}" row and column of d(x,y) are equal to some "average" value. This average value must be stored separately. For convenience, we can take this to be the median value of the first row of the image.) In the case of multiband images, each band is transformed separately according to equation (1) above.

The entropy (Shannon[9]) of an image is obviously dependent on the definition of the "source alphabet" and statistics of the reception of symbols from it. In the one extreme, the image could be considered to be obtained from a binary source (i.e., as a serial bit stream). In the other extreme, the source could be an "image generator" which produces images of a given size and the given image would then be regarded as an instance from the ensemble of all possible images (i.e., a single symbol from a very large alphabet!). In practical applications, the source alphabet is defined to consist of single or multiband n-bit pixel values (e.g., for 7-band Landsat TM data the source alphabet could either be all possible 8-bit pixel values or all possible 56-bit pixel values).

The zeroth order entropy, H0, of an image, d, is given by

$$H_0 \ (d) = -\sum_{i=1}^{\beta} P_i \ (d) \ x \ \log_2 \ (P_i \ (d)) \qquad (2)$$

where Pi(d) is the probability of a pixel in image, d, having value i. β = 2, where b is the number of bits per pixel in the image. We calculate LCP of image d(x,y) by first finding the difference image d'(x,y) through the process defined by equation (1). We then estimate the pixel value probabilities, Pi(d'), from the histogram of d'(x,y) and calculate the zeroth order entropy, H_0(d'), through equation (2). To complete the definition of the LCP we note that in order to decode the compressed image reconstructing the original image we need to know the code used to encode the image. We assume that a variable length Huffman code is used to encode the image to achieve close to ideal. Thus, we define our LCP as

$$LCP = \frac{bxN}{H_0xN + B_c} \qquad (3)$$

where b is the number of bits per pixel, N is the total number of pixels in the image, B_C is the total number of bits needed to describe the Huffman Code, and H_O is the zeroth order entropy of the difference image, d'(x,y).

In the case of multiband images, the encoding can be done by treating each band separately (band-by-band compression) or by treating them all together (across-band compression).

From equation (3), we see that as N becomes sufficiently large, the overhead of storing the code becomes negligible. Since the variable length Huffman code is uniquely determined by the ranking of frequencies of the grey levels in the image, a means of storing the code is to store the rank order table derived from the image histogram. In the case of a single band image with b-bit pixels, the number of bits required to store the compression code is bounded by

$$B_C \leq b + 4x2^b x(b+2) \qquad (4)$$

assuming b bits to store the median of the original image d(x,y) and (b+2) bits per entry in the rank order table of the transformed image d'(x,y) with $4x2^b$ possible entries.

In the case of multiband image taken together (across-band compression), the number of possible entries in the histogram becomes quite large and, if we store all entries (including those with zero frequencies) the overhead for code storage becomes considerable. One way around this is to store the histogram as a paired table, with entries (viz, multiband image value, frequency or rank order) only when the frequency is nonzero.

However, unique vector counting experiments (Wharton[11]) have shown that for moderately sized (say 512x512) images, the number of entries in the histogram of a 7-band TM image are comparable to the image size itself. In these cases there is nothing to be gained by attempting across-band compression as defined above.

Error Measures

A widely used method for evaluating the quality of a compressed and reconstructed image relative to the original image is the Mean Squared Error (MSE). The MSE of band "i" of a multiband image is defined as

$$MSE_i = E[(D_i - D^r{}_i)^2] \simeq 1/N-1 \sum_{p=1}^{N} (D_{ip} - D^r{}_{ip})^2 \qquad (5)$$

where D_i and $D^r{}_i$ are the data values of the ith band of the original and reconstructed images, respectively; D_{ip} and $D^r{}_{ip}$ are the values of the pth pixel of the ith band of the original and reconstructed images, respectively; E denotes the expected value; and N is the total number of pixels in the image.

The MSE_i as defined above is a single-band error measure. One could define a multiband MSE by simply summing the MSE_i over the bands. However, this definition does not account for the differences in variance between individual bands, and the values that would be obtained do not correspond to a direct conceptual notion of error. We prefer an error measure we call the multiband Root Normalized RNMSE, which we define as follows:

$$RNMSE = 1/m \sum_{i=1}^{m} \sqrt{MSE_i/VAR_i} \qquad (6)$$

where VAR_i is the variance of the ith band and m is the number of bands in the multiband image. In addition to accounting for the differences in variance between individual bands, the RNMSE carries an intuitive interpretation: The RNMSE is the band average of the single-band RNMSE, which can be regarded as the mean deviation of a reconstructed image pixel value from the corresponding original image pixel value per standard deviation of the band.

Spatially Constrained Clustering (SCC)

SCC is an iterative parallel segmentation approach that performs the "globally best merge" among spatially adjacent regions at each iteration. The globally best merge is the merge with the best similarity criterion value over all pairs of spatially adjacent regions. As implemented here, the SCC algorithm starts by initializing each pixel as a separate region. The globally best pair of regions are then merged at each iteration. The algorithm is considered to have converged when either a desired number of regions remain, or when no pair of adjacent regions is similar enough to be merged according to a predefined bound on the similarity criterion. A key aspect of any region growing approach is the similarity criterion used to determine whether or not a region should grow by merging with a neighboring region or pixel. The best similarity criterion depends upon the application. To fully explore the utility of the general SCC approach, we need to devise and test several different similarity criteria for different types of scientific image data and for various analysis procedures performed on each type of scientific image data. In the experiments reported here, the similarity criterion used is based on minimizing variance normalized mean squared error.

In the previous section we defined the mean squared error for band "i", MSE_i (see equation 5). The variance normalized mean squared error for band "i" ($NMSE_i$) is defined as

$$NMSE_i = \frac{MSE_i}{VAR_i} \qquad (7)$$

where VAR_i is the variance of band "i", as before. The similarity criterion used in our tests is the $MAX(\Delta NMSE_i)$ for each pair of spatially adjacent regions, where the maximum is taken over all bands

($1 \leq i \leq m$). For a particular pair of spatially adjacent regions, $\Delta NMSE_i$ is the change in $NMSE_i$ when the pair of regions is merged and the reconstructed image is formed by substituting the mean grey level of each region for the grey level for each pixel in the region. The globally best merge is then the pair of regions, out of all spatially adjacent regions, that minimizes the similarity criterion.

The change in $NMSE_i$, or $\Delta NMSE_i$, is calculated as follows. Define

$$\Delta NMSE_i = \frac{MSE^c{}_i - MSE_i}{VAR_i} \qquad (8)$$

where $MSE^c{}_i$ is the mean squared error when regions j and k are merged, while MSE_i is the mean squared error before regions j and k are merged. Using the definitions of MSE_i, and the region mean, it is easy to derive a more fundamental version of equation (8), viz

$$\Delta NMSE_i = \frac{n_j\ (\bar{D}_{ij} - \bar{D}_{ijk})^2 + n_k(\bar{D}_{ik} - \bar{D}_{ijk})^2}{(N - 1) VAR_i} \qquad (9)$$

where n_j and n_k are the number of points in regions j and k, respectively, before combining, and N is the number of points in the image. D_{ij} and D_{ik} are the mean values of band i for regions j and k, respectively, before combining, and D_{ijk} is the mean value of band i for the region that would result from combining regions j and k.

We have implemented the SCC algorithm on the Massively Parallel Processor (MPP) at the NASA Goddard Space Flight Center. The MPP is a Single Instruction, Multiple Data stream (SIMD) computer containing 16,384 bit serial microprocessors logically connected in a 128-by-128 mesh array with each microprocessor have direct data transfer interconnections with its four nearest neighbors. With this massively parallel architecture, the MPP is capable of billions of operations per second.

Block Cluster Coding

The SCC segmentation is encoded for storage or transmission by a block encoding technique. The current implementation of the SCC algorithm performs segmentations on relatively small blocks of image data (from 32x32 to 128x128) because of memory limitations on the MPP. (NOTE: This limitation has been lifted in a more recent implementation in which image data and intermediate results are stored temporarily in a "staging buffer memory.") This restricts the block sizes to be used for encoding to sizes that can be evenly divided into the SCC segmentation block size. The SCC segmentation block size for the 7-band Landsat TM data used in this study was 42-by-42 pixels. This restricted the encoding block sizes to 42-by-42, 21-by-21, 14-by-14, 7-by-7, 6-by-6, 3-by-3, or 2-by-2 pixels. (The 6-by-6, 3-by-3 and 2-by-2 block sizes were not used because encoding becomes inefficient at very small block sizes.) The optimal encoding block size must be determined empirically for each application.
In performing the block cluster coding, two files are created. The region labels in each block are renumbered to use the minimal number of bits and stored as the region map file, and the mean vectors for each region in each block are stored in a region feature file. The region map file is further losslessly compressed by an appropriate method. A method we found to be effective is run-length coding along bidirectional scan lines (odd lines scanned left to right, even lines scanned right to left) with maximum run length equal to the number of samples in each line of the coding block.

This compression scheme of segmentation followed by block cluster coding was inspired by the Cluster Compression Algorithm (CCA) developed by Hilbert(2). The main difference between CCA and our approach is the segmentation algorithm used to define the regions.

EXPERIMENTAL EVALUATION

In this section, we describe the data sets used for the evaluation of data compression procedure, the quality criteria considered, and the experimental results.

The Data Sets

A 468-by-368, 7-band subset of a Landsat TM image is one of the data sets used in this study. The subset is registered to a U.S. Geological Survey topographic map of the Ridgely Quadrangle (7.5 minute quad sheet) in Maryland. This particular image subset was chosen because the data had a sufficient variety of classes and it had a digitized ground truth map that was registered and rectified to the topographic map. In addition, this data set was used in an earlier data compression study (Ramapriyan[7]).

The other data set used in this study is a 486-by-1968 section of a 5-band Nimbus-7 CZCS image. This image was collected on October 25, 1980 over a section of the equatorial Pacific Ocean, and contained substantial numbers of scattered clouds in the western half of the image, and some very heavy clouds in the far eastern quarter of the image. The remaining quarter of the image was almost completely cloud free. The CZCS data contrasts sharply with the TM data set in that, except for the clouds, the CZCS has no obvious spatial features, while the TM data set has numerous, very obvious, spatial features. The CZCS image has no "ground truth" file.

Quality Criteria

The complexity of each dataset is measured using the band-by-band LCP. We measured the effects of data compression on both the TM and CZCS data sets by calculating the RNMSE. However, the RNMSE does not necessarily measure how much scientifically relevant information is retained in the compressed image. For the TM data we used the scientifically relevant quality measure of classification accuracy using two different classification approaches. We have as yet not developed a scientifically relevant quality measure for the CZCS data.

The original and SCC compressed and reconstructed TM images were classified using the Maximum Likelihood Classifier (MLC) (Swain and Davis[10]). For comparison, the original TM image was also classified using the Supervised Extraction and Classification of Homogeneous Objects (SECHO) classifier (Kettig and Landgrebe[6]). The spectral classes required by these classifiers were selected from clusters generated by the ISOCLASS algorithm (Kan, Holley and Parker[4]) from NASA GSFC's Land Analysis System (LAS). The ISOCLASS algorithm was used on the entire TM test image to produce 64 clusters. It was also used on areas of the image with a high proportion of the residential and water/marshland ground cover classes to produce 16 and 32 additional clusters, respectively. The resulting 112 clusters were reduced to 31 spectral classes based on visual inspection and suppression of the most overlapping classes (within each ground cover class). For our tests, four informational classes were de-fined: Water/Marshland, Forest, Residential, and Agricultural/Domestic Grass. The means and covariance matrices of the spectral classes were then used to perform "supervised" classifications of the image using both MLC and SECHO. The classified images were mapped into the four information classes, and the resulting label images were compared pixel by pixel with the ground truth label image to obtain the classification accuracies.

Experimental Results

The LCPs for each of the seven bands of the TM test image are given in Table 1. The average LCP across all bands is 2.76. This means that the test image could be compressed by a factor of at least 2.76 (but probably not much more) without loss of any information. The large value of the average LCP is primarily due to the large LCP for band 6 (6.47), the low resolution thermal band. The average LCP over the six reflective (full resolution) bands is 2.14.

In Table 2 we show compression factors for the TM test data set resulting from the SCC segmentation followed by encoding with blocks of various sizes and run-length coding. The compression factor tends to peak for an encoding block of size 21x21 or 42x42, with block size 14x14 trailing close behind. The "Threshold" shown in the table is the maximum NMSEi allowed in the SCC algorithm.

Table 1. Lossless Compression Potential (LCP) of the 7-Band Thematic Mapper Test Image.

								Band
Band	1	2	3	4	5	6	7	Ave.
LCP	2.11	2.66	2.28	2.09	1.72	6.47	1.96	2.76

Table 2. Compression Factors for Varying Encoding Block Size for the 7-Band Thematic Mapper Test Image.

Threshold	Encoding Block	CF	CF/ LCP*
0.1	42x42	6.21	2.25
"	21x21	6.36	2.30
"	14x14	6.13	2.22
"	7x7	5.02	1.82
0.2	42x42	13.6	4.93
"	21x21	13.6	4.93
"	14x14	12.5	4.53
"	7x7	8.77	3.18
0.3	42x42	23.3	8.44
"	21x21	22.5	8.15
"	14x14	19.8	7.17
"	7x7	12.2	4.42

*LCP here is the Band Average LCP = 2.76.

The RNMSE image quality measure for three $NMSE_i$ thresholds is given in Table 3 for the TM test data set. Classification accuracy evaluations are given in Table 4 for the MLC algorithm for the original and three cases of compressed TM data. For comparison, the classification accuracy is also given for the SECHO classifier on the original data. As can be seen by inspecting the accuracy figures in Table 4, for the MLC algorithm the classification accuracies are consistently as good or better for the compressed data than they were for the original data. For most cases, the MLC classification accuracies are better for the compressed data than the classification accuracies for the SECHO classifier on the original data. In fact, the classification accuracies obtained by running the MLC algorithm on the data that was compressed by a factor of 23.3 are consistently better than the accuracies obtained by running either the MLC algorithm or the SECHO classification algorithm on the original data! We hypothesize that the SCC segmentation is behaving like a more sophisticated homogeneous object extraction procedure than that used in the SECHO classification algorithm.

Table 3. Reconstructed Image Quality
for the 7-Band Thematic Mapper Test Image.

Threshold	RNMSE	CF*	CF/ LCP*
0.1	0.23	6.36	2.30
0.2	0.32	13.6	4.93
0.3	0.38	23.3	8.44

*This is the maximum CF and CF/LCP over various encoding block sizes.

Table 4. MLC and SECHO Classification Accuracy Comparisons for the 7-Band Thematic Mapper Test Image (% correct classification).

Class	MLC on Original Image	MLC on CF*= 6.36	MLC on CF*= 13.6	MLC on CF*= 23.3	SECHO on Original Image
Water/Marsh	58.6	58.6	61.0	61.8	58.4
Forest	67.3	68.3	69.2	68.7	65.5
Residential	54.4	60.2	60.5	70.8	67.5
Ag./Dom. Grasses	84.9	85.9	86.6	85.9	83.5
Overall	80.0	81.1	81.8	81.3	78.7

*This is the maximum CF over the various encoding block sizes.

A subjective evaluation of the reconstructed TM images shows that areas in the original image which are relatively homogeneous, but not necessarily uniform, become completely uniform in the reconstructed images. Low contrast spatial features are often lost in the reconstructed images, but the higher contrast spatial features are retained very precisely. Even very small spatial features are retained if they have sufficient contrast relative to the surrounding area. Further experiments are needed to verify whether the SCC compression approach effectively retains all relevant scientific information in Landsat TM data. The above results seem to indicate, however, that this compression approach retains much of what would seem to be the relevant scientific information.

The LCPs for each of the five bands of the CZCS test image are given in Tables 5a and 5b. The average LCP across all bands of the entire image is 1.89. However, the LCP across all bands for a 486-by-504 pixel cloud-free section of data is 3.45.

Table 5a. Lossless Compression Potential (LCP) of the 5-Band Coastal Zone Color Scanner Test Image (Full scene - 486 lines by 1968 columns).

Band	1	2	3	4	5	Band Ave.
LCP	1.90	1.78	1.71	1.50	2.57	1.89

Table 5b. Lossless Compression Potential (LCP) of the 5-Band Coastal Zone Color Scanner Test Image (Cloud free section - 486 lines by 504 columns).

Band	1	2	3	4	5	Band Ave.
LCP	3.00	3.18	3.04	2.35	5.69	3.45

In Tables 6a and 6b (full scene and cloud-free section, respectively) we show compression factors for the CZCS test data set resulting from the SCC segmentation followed by encoding with blocks of various sizes and run-length coding. The compression factor tends to peak for an encoding block of size 21x21 or 42x42, with block size 14x14 trailing close behind.

The RNMSE image quality measure is given in Table 7 for the CZCS test data set. These results are inconclusive, but a visual inspection of the mean images produced by the SCC algorithm shows that the algorithm behaves poorly only in the vicinity of the clouds. The very high variance of the clouds cause the algorithm to segment very coarsely in the vicinity of clouds compared to elsewhere in the image. Modifying the algorithm to use the variance of the whole image rather than just the variance of the individual segmentation blocks in calculating the variance normalized mean

Table 6a. Compression Factors for Varying Encoding Block Size for the 5-Band Coastal Zone Color Scanner Test Image (Full scene - 486 lines by 1968 columns).

	Encoding		CF/
Threshold	Block	CF	LCP*
0.3	42x42	8.53	4.51
"	21x21	8.94	4.73
"	14x14	8.66	4.58
"	7x7	6.93	3.67

*LCP here is the Band Average LCP = 1.89.

Table 6b. Compression Factors for Varying Encoding Block Size for the 5-Band Coastal Zone Color Scanner Test Image (Cloud free section - 486 lines by 504 columns).

Threshold	Encoding Block	CF	CF/ LCP*
0.3	42x42	3.92	1.14
"	21x21	4.14	1.20
"	14x14	4.11	1.19
"	7x7	3.64	1.06
0.5	42x42	9.86	2.86
"	21x21	10.5	3.04
"	14x14	10.9	3.16
"	7x7	7.58	2.20
0.7	42x42	24.2	7.01
"	21x21	23.7	6.87
"	14x14	21.0	6.09
"	7x7	13.0	3.77

*LCP here is the Band Average LCP = 3.45.

Table 7. Reconstructed Image Quality for the 5-Band Coastal Zone Color Scanner Test Image

Scene	Threshold	RNMSE	CF*	CF/ LCP*
Full scene	0.3	0.28	8.94	4.73
Cloud-free sec.	0.3	0.11	4.14	1.20
Cloud-free sec.	0.5	0.14	10.9	3.16
Cloud-free sec.	0.7	0.17	24.2	7.01

*This is the maximum CF or CF/LCP over the encoding block sizes.

squared error (equation 7), should improve the behavior of the algorithm. However, going back to our original premise of tailoring our compression approach to the characteristics of the data, we question the utility of pursuing this approach further for CZCS data.

Except in the vicinity of clouds and land masses, the CZCS image data generally has very little spatially variability. In the case of ocean images, there are no distinct boundaries as seen in the case of the land images (e.g., between a forested area and an agricultural field as found in TM image data). Since the forte of the SCC approach is the preservation of boundaries between contrasting regions, it may make little sense to apply it to data, such as CZCS data, where such boundaries aren't important. The only contrasting boundaries found in CZCS data are between clouds and ocean, and land and ocean. The users of CZCS data routinely mask out and discard cloudy data and data collected over land using simple thresholding schemes. A more appropriate compression approach may be to mask out the cloudy data and data collected over land in the same way done routinely now by the users of the data and use some variation on run-length encoding to compress the remaining data.

CONCLUDING REMARKS

Given the resolutions and data rates expected from the remote sensors to be flown during the next two decades, it will be necessary to consider both lossless and lossy data compression techniques to keep the transmitted data rates and archived data volumes within manageable limits. Lossy techniques, wherein the raw data bits cannot be exactly reconstructed, require careful studies in coordination with scientific users to determine whether most of relevant information for a given application is retained. Several such studies are needed in selected disciplines and application areas. In this paper, we have presented one such study using a compression technique which can be a precursor to analysis and information extraction. The compression technique is based on SCC and subsequent local encoding of regions. This is an adaptive vector quantizer where very short codebooks are needed and are developed "on the fly" for local rectangular regions.

Since the SCC is a segmentation technique, it is a very useful precursor to the analysis of image data with significant amount of detail. The algorithm can be controlled with a single parameter to obtain different degrees of segmentation (retaining different levels of detail) and corresponding compression ratios. In our case study, we have explored the use of the SCC with two types of data. The first, a land image from the Landsat TM, has considerable spatial detail while the second, an ocean image from the CZCS has no recognizable features except clouds (which are usually suppressed in performing any analyses).

For the case of the TM data, we found that land cover classification accuracies are higher with compressed data than with raw data even up to compression ratios over 20. This agrees with results from earlier studies with other compression techniques such as the Cluster Coding Algorithm (CCA). Further experiments are needed to verify whether all relevant scientific information is retained by such compression techniques. However, the present study confirms that for land cover classification applications significantly compressed data can be used directly, and in many cases, more usefully than raw data. In the case of CZCS data, the image distortion measures and subjective image evaluation show that compression ratios of 4 to 24 can be achieved with relatively small distortions. Further experiments on derived geophysical parameter data are needed to examine the impact of compression on the analysis of CZCS data. However, given the nature of the CZCS data it is probably more fruitful to consider other compression techniques such as linear prediction and run length encoding.

REFERENCES

1) Chen, T.M., D.H. Staelin and R.B. Arps, "Information Content Analysis of Landsat Image Data for Compression,", IEEE Transactions on Geoscience and Remote Sensing, vol. GE-25, #4, pp 499-501, July 1987.

2) Hilbert, E. E., "Cluster Compression Algorithm: A Joint Clustering/Data Compression Concept", JPL Publication 77-43, Jet Propulsion Laboratory, Pasadena, CA, 1977.

3) Jain, A. K., "Image Data Compression: A Review", Proc, IEEE, Vol. 69, pp. 349-389, 1981.

4) Kan, E. P., W. A. Holley and H. D. Parker, Jr., "The JSC Clustering Program ISOCLS and its Applications", Proceedings of the 1973 Machine Processing of Remotely Sensed Data Symposium, (IEEE Catalog No. 73 CH 0834-2 GE), Oct. 16-18, 1973, pp. 4B-36 thru 4B-45.

5) Kauth, R.J., et al, "BLOB: An Unsupervised Clustering Approach To Spatial Preprocessing of MSS Imagery", Proceedings of 11th International Symposium of Remote Sensing of the Environment, Environmental Institute of Ann Arbor, Michigan, 1977.

6) Kettig, R. L. and D. A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects", IEEE Trans. on Geoscience Electronics, pp. 19-26, 1976.

7) Ramapriyan, H. K., J. C. Tilton and E. J. Seiler, "Impact of Data Compression on Spectral/Spatial Classification of Remotely Sensed Data", Advances in Remote Sensing Retrieval Method, Deepak, Fleming and Chahine, Eds., pp. 687-706, 1985.

8) Rosenfeld, A. and A. C. Kak, "Digital Picture Processing", 2nd Ed., Acad. Press, NY, pp. 181-182, 1982.

9) Shannon, C. E., "The Mathematical Theory of Communication", Bell System Technical Journal, Parts I and II, pp. 379-423, 623-656, 1948.

10) Swain, P. H. and S. M. Davis (Eds.), "Remote Sensing: the Quantitative Approach", McGraw-Hill, NY, 1978.

11) Wharton, S. W., "An Algorithm for Computing the Number of Distinct Spectral Vectors in Thematic Mapper Data", IEEE Trans. on Geoscience and Remote Sensing, pp. 294-302, 1985.

DATA COMPRESSION

ALGORITHM R&D

JIM STORER

BRANDEIS UNIVERSITY

SESSION COORDINATOR

A RECURSIVE TECHNIQUE FOR ADAPTIVE VECTOR QUANTIZATION

Robert A. Lindsay
Unisys Corporation

ABSTRACT

Vector Quantization (VQ) is fast becoming an accepted, if not preferred method for image compression. VQ performs well when compressing all types of imagery including Video, Electro-Optical (EO), Infrared (IR), Synthetic Aperture Radar (SAR), Multi-Spectral (MS), and digital map data. The only requirement is to change the codebook to switch the compressor from one image sensor to another. However, codebooks can be difficult to design because data may not be available or may not accurately represent the pdf. This stimulates the need for an algorithm that can simultaneously design a codebook while vector quantizing the data.

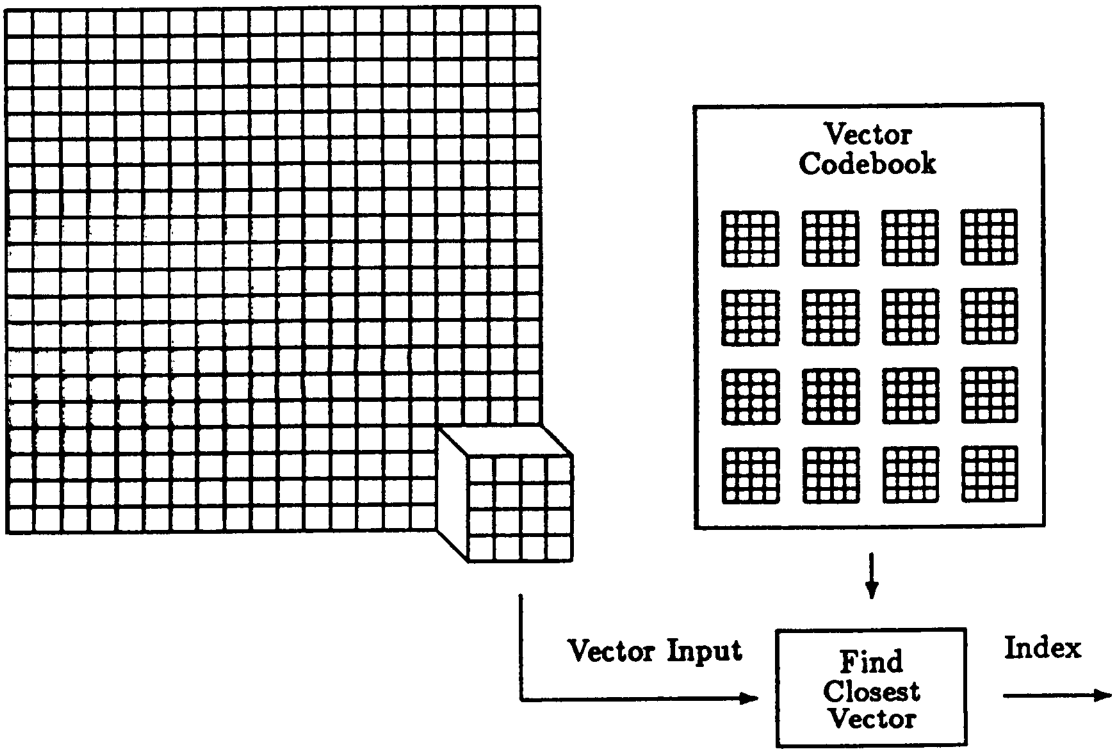
There are several approaches for designing codebooks for a vector quantizer. The most common algorithm being the LBG or generalized Lloyd. Entries in the codebook represent the centroid of the data that is associated with a respective Voroni region. A quantizer is uniquely defined by the codebook centroids and the distortion metric. The LBG algorithm is used to minimize the overall distortion of the quantizer by iteratively moving the centroids and computing the new distortion until the quantizer converges on a local minimum. Previous implementations of the LBG algorithm compute the centroid by adding all the vectors in the Voroni region and then dividing by the number of vectors. This is done iteratively on a sample of source data referred to as a training sequence.

Adaptive Vector Quantization is a procedure that simultaneously designs codebooks as the data is being encoded or quantized. This is done by computing the centroid as a recursive moving average where the

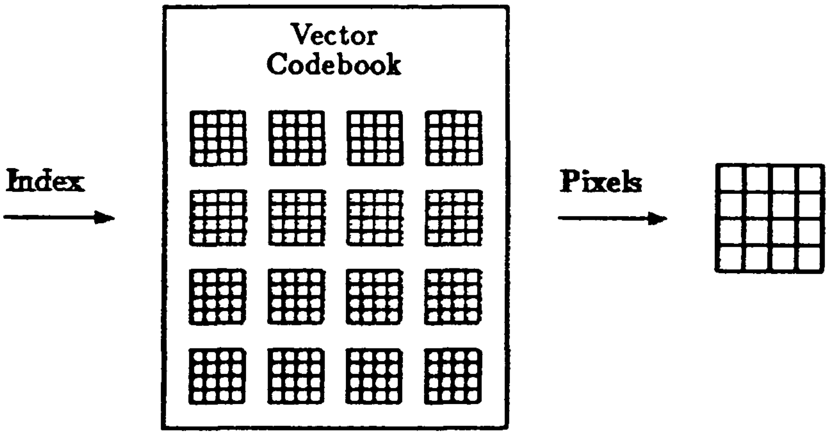
centroids move after every vector is encoded. When computing the centroid of a fixed set of vectors the resultant centroid is identical to the previous centroid calculation. This method of centroid calculation can be easily combined with VQ encoding techniques. The defined quantizer changes after every encoded vector by recursively updating the centroid of minimum distance which is the selected by the encoder. Since the quantizer is changing definition or states after every encoded vector, the decoder must now receive updates to the codebook. This is done as side information by multiplexing bits into the compressed source data. It is important to note that the quantizer converges in much the same way as the LBG algorithm converges. For stationary data sources the centroids will become fixed and the side information will not be necessary. For non-stationary sources the side information can be used to allow the quantizer to adapt to the data, thereby providing an Adaptive Vector Quantizer. Important issues to consider are the rate of convergence, start-up distortion or rate overhead, and tracking non-stationary sources. These issues will be addressed in a forthcoming publication.

ACKNOWLEDGEMENTS

This work was partially supported by Unisys Corporation
through the University of Utah Center for Communications Research.



Vector Quantization Encoding



Vector Quantization Decoding

Present Implementation of VQ

- Acquire data from sensor
- Design a codebook
- Implement a search technique

Acquire Data from the sensor

- Expensive
- Classified
- Not possible
- Poor representative
  (start over)

Codebook Design

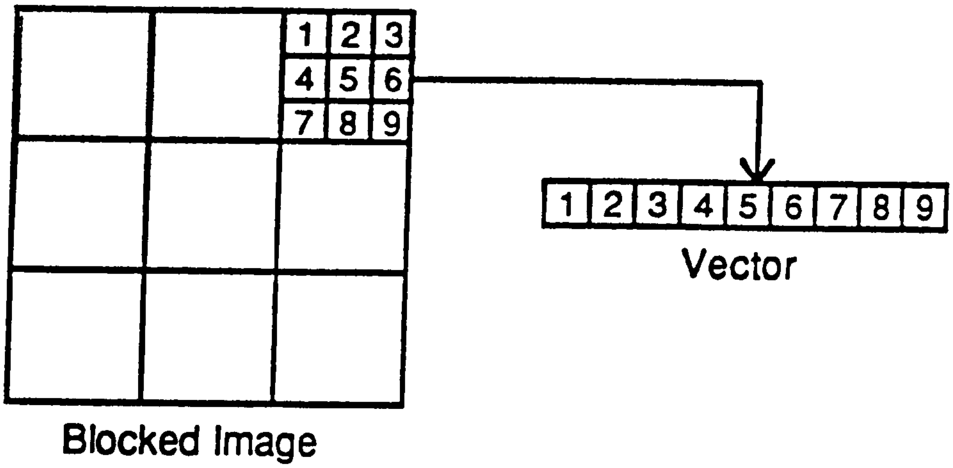
- Exhaustive Search
- Generalized Lloyd (LBG)
- K-means
- Simulated Annealing
- Pairwise Nearest Neighbor (PNN)

One Dimensional Lloyd's Algorithm

- Determine PDF of source
- Solve Equations
  - $x_j = \frac{y_j + y_{j-1}}{2} \qquad j = 2, ..., N$
  - $\int_{x_j}^{x_{j+1}} (x - y_j) p(x) dx = 0 \qquad j = 1, ..., N$

Generalized Lloyd using a training sequence

- Acquire training sequence
  (lots of samples of source data)
- Create vectors by grouping samples
  (maximize correlation)



- **Design codebook**

1. **Initialize codebook**

   Place a set of quantization points in the vector space

2. **Encode training sequence**

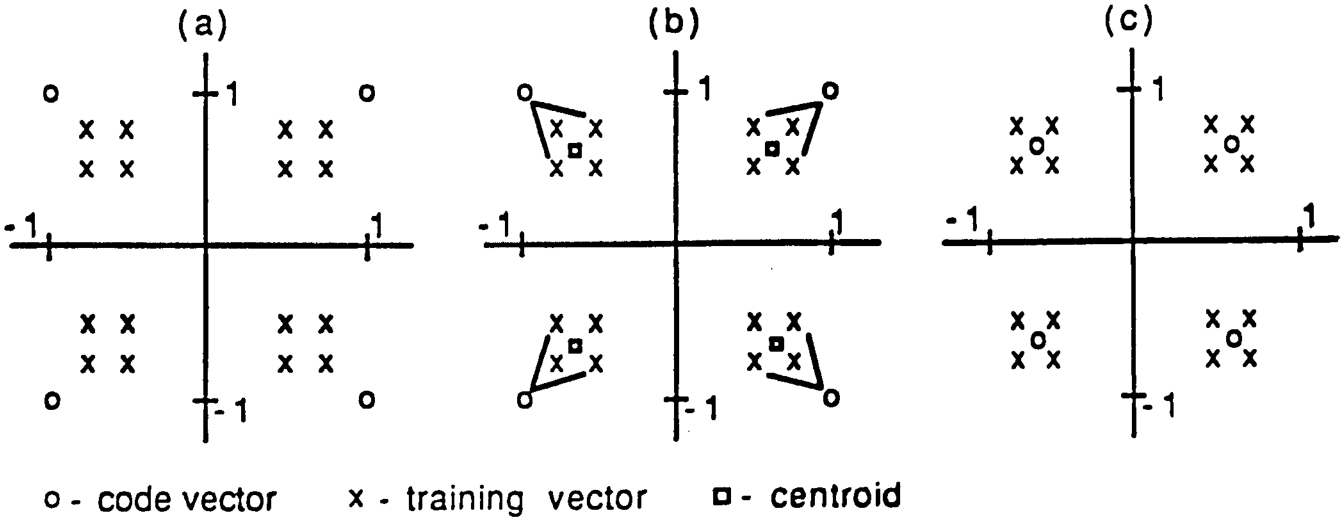
   Assign each vector from the training sequence to the closest quantization point

3. **Reassign codebook**

   Compute the centroid of each set of training sequence vectors assigned to a codebook vector and reassign the codebook to be these new centroids

4. **Iterate 2 and 3**

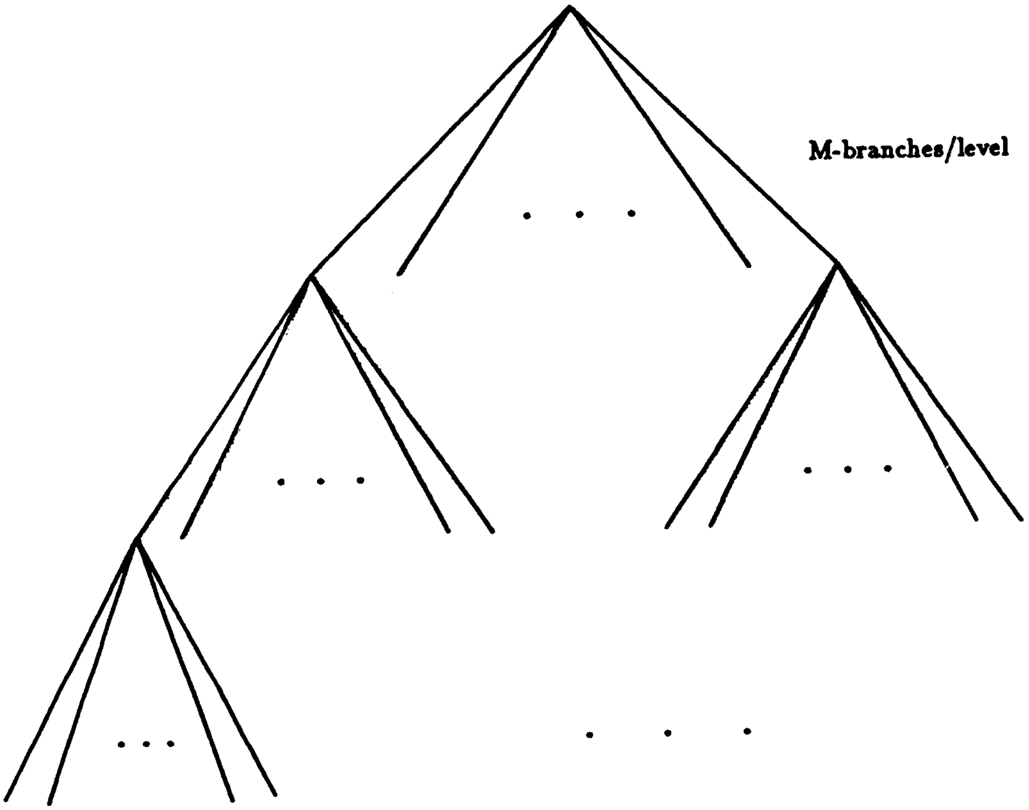
   Iterate until no change (or minor change) to the overall distortion



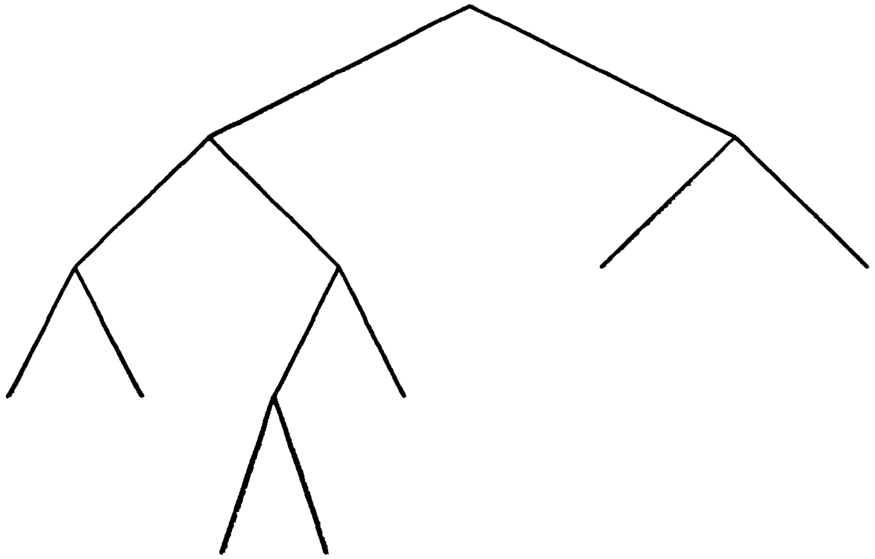
Encoding the Source Vectors
Using Full Search

- Compute the quantization error (distortion) between the source vector and each vector in the codebook
- Replace the source vector with the index to the vector of minimum distortion

Tree



NON-UNIFORM BINARY TREE SEARCH



Each level computes $\vec{s} \cdot \vec{v}(j) \lesseqgtr T(j)$ where

- $\vec{v}(j) = \vec{C}_2(j) - \vec{C}_1(j)$
- $\vec{T}(j) = \dfrac{\|\vec{C}_2(j)\|^2 - \|\vec{C}_1(j)\|^2}{2}$

Adaptive Vector Quantization

- Combines principles of codebook design with encoding
- Requires no source samples of data to start (training sequence)
- Removes logistics problem of changing codebooks

Recursive Codebook Design

- Initialize codebook vectors
- Encode a source vector as before by looking at each entry in the codebook and choosing entry of minimum distortion
- Update the codebook vector after every encoded source vector
- Send Δchange as side information

Changing Codebook Values

- Codebook entries are the centroids of the Voroni region
- Centroid computation

  1. $C = \frac{1}{N} \Sigma_{i=1}^{N} x_i$
  2. $C_n = \frac{n-1}{n} C_{n-1} + \frac{1}{n} x_n \qquad n = 0, 1, ..., N$

- Codebook is a set of changing centroids

$$y_{n_1} = \frac{n_1 - 1}{n_1} y_{n_1 - 1} + \frac{1}{n_1} x_{n_1} \qquad n_1 = 0, 1, ...$$
$$y_{n_2} = \frac{n_2 - 1}{n_2} y_{n_2 - 1} + \frac{1}{n_2} x_{n_2} \qquad n_2 = 0, 1, ...$$
$$\vdots$$
$$y_{n_M} = \frac{n_M - 1}{n_M} y_{n_M - 1} + \frac{1}{n_M} x_{n_M} \qquad n_M = 0, 1, ...$$

- Centroids converge in much the same way as the Generalized Lloyd algorithm

Things to Consider

- Start-up
  - Increase rate
  - Increase distortion
  - Reset
- Convergence
- Register overflow
  - N counts
  - α divides
- Overhead for side information
- Performance
- Stationarity of source

FRACTAL IMAGE COMPRESSION

Michael F. Barnsley and Alan D. Sloan
Mathematics, Georgia Tech

ABSTRACT

Fractals are geometric or data structures which do not simplify under magnification. Fractal Image Compression is a technique which associates a fractal to an image. On the one hand, the fractal can be described in terms of a few succinct rules, while on the other, the fractal contains much or all of the image information. Since the rules are described with less bits of data than the image, compression results.

Data compression with fractals is an approach to reach high compression ratios for large data streams related to images. The high compression ratios are attained at a cost of large amounts of computation. Both lossless and lossy modes are supported by the technique. The technique is stable in that small errors in codes lead to small errors in image data. Applications to the NASA mission are discussed.

OVERVIEW

Fractals are geometric or data structures which do not simplify under magnification. Fractal Image Compression is a technique which associates a fractal to an image. On the one hand, the fractal can be described in terms of a few succinct rules, while on the other, the fractal contains much or all of the image information. Since the rules are described with less bits of data than the image, compression results.

Fractal image compression is a computationally intensive technique.

However, the computations required are mainly multiplications and accumulations and iterative in nature. The rules consist of low dimension matrix transformations. Therefore, high speed hardware implementations are possible. A hardware decoder was demonstrated in October, 1987. This prototype device can decode 256 x 256 x 8bits/pixel images at a rate of several frames per second. It demonstrates the feasibility of higher performance decoders.

The Collage Theorem, described in the next section, provides the connection between rules and images. It allows for the complete control of the fidelity of the encoded image. Compression ratios in the lossy mode are typically much higher than in the lossless mode. The association between fractals and images is a continuous one in the following sense: small changes in the matrices produce small changes in images.

The observation that a simple set of rules can produce a complex image began with abstract fractal pictures known as Julia sets. The Georgia Tech mathematics research team set out to explore the limits of this observation. The exploration took the form of a search for a solution to an "inverse" problem (e.g., given an image, find the rules which encode it as a fractal). The Collage Theorem is a remarkable solution to this inverse problem. High resolution, color images have been encoded in several thousand bytes.

Basic research in this technique at Georgia Tech and other universities has been supported by DARPA, AFOSR, NSF and ONR. Georgia Tech was specifically funded under the Applied and Computational Mathematics Program of DARPA to investigate automation of this technique using simulated thermal annealing algorithms. While basic research continues, a number of corporations are investigating applications of this technique. In particular, Iterated Systems, Inc. was formed to commercialize this technology.

THE TECHNIQUE

Fractal Image Compression Codes

The code for an image which has been subjected to fractal image compression consists of an iterated function system (ifs). An ifs is composed of affine transformations and probabilities. In two dimensions, an affine transformation, T, takes the form

$$T(x,y) = (x',y')$$

where $x' = Ax + By + C$

and $y' = Dx + Ey + F$

The six coefficients A,B,C,D,E,F define T and must be specified in log(resolution) bits. For example, 100 affine transformations on a 1024 x 1024 screen require 100*6*10 bits. Affine transformations consists of scalings, rotations and translations and so have a geometrical interpretation. The affine transformations which appear in ifs codes should be contractive in the following sense. If $|P-Q|$ denotes the Euclidean distance between two points, P and Q, and if T is a contractive transformation, then

$$|T(P) - T(Q)| < s * |P-Q|$$

where the contractivity factor $s < 1$.

The probabilities form a linkage matrix used in the decoding process. In the most common usage to date, only one probability is specified for each transformation and these may be specified in log(number of colors) bits. A color look-up table must also be specified for use in the decoder. Typically, several color values are given and linear interpolation is used to generate intermediate colors. In typical examples this can be done in less than log(number of colors) bits/map. Thus, encoding a 1024 x 1024 x 9 bits/pixel image in N maps results in a code of length 72* N bits. As an example, choosing N = 100 results in a length of 7200 bits. The original image is 1024 * 1024 * 9 bits so the compression ratio exceeds 1000 : 1.

Some ifs codes are given on Tables 1-4.

Hausdorff Image Distance

Precise statements concerning fractal image encoding and decoding refer to the Hausdorff distance between images. Fix a screen, S, consisting of R rows with P pixels/row. A monochrome image (1 bit/pixel) is simply a collection, I, of pixel sites on the screen which are illuminated. The distance from a screen location a to an image B is defined as the closest Euclidean distance. That is,

$$d(a,b) = \text{minimum } \{ |a-b| : \text{for } b \text{ in } B\}.$$

The distance from an image A to the image B is given by

$$d(A,B) = \text{maximum } \{ d(a,B) : \text{for } a \text{ in } A\}.$$

This max-min type distance function is not symmetric. That is, d(A,B) may not coincide with d(B,A). See Figure 2.

The Hausdorff distance between A and B is

$$H(A,B) = \text{maximum } \{ d(A,B), d(B,A)\}.$$

If the Hausdorff distance between two images is zero, then the images are identical. If the Hausdorff distance is less than the resolution of the screen, the two images are indistinguishable.

The Hausdorff distance definition may be generalized to color and grey-scale images by viewing the color information as a third coordinate. From this point of view, images are surfaces and the

Hausdorff metric is used to measure distances between surfaces.

The Decoder

An ifs code can be thought of as an image processing operation. Let C be an ifs code consisting of the affine transformations T1, T2, ..., Tn. If R is any image, then Ti(R) means the image under Ti of all the points in R. C(R) is then defined by

$$C(R) = T1(R) \cup T2(R) \cup \ldots \cup Tn(R).$$

As an example, let the image R consist of one point and suppose the code consists of two transformations. Then typically, C(R) will consist of two points. Starting now with C(R), C(C(R)) will typically consist of four points. Since, however, the affine transformations in an ifs code are contractions, a transformation may coalesce several points into a single point.

Associated to every ifs code is a unique set called the attractor of the code. The attractor A = attr(C) of the ifs code C is defined as the only set with the property that

$$A = T1(A) \cup T2(A) \cup \ldots \cup Tn(A).$$

The maximum of all the contractivity factors of the affine transformations in an ifs code is the called the contractivity factor of the code.

Let A(0) be any non-zero subset. Inductively define A(i) = C(A(i-1)). Then the sequence A(0), A(1), A(2) converges to the attractor A of the ifs code C, in the Hausdorff metric. That is, as i gets large H(A(i), A) becomes small. For typical examples, the Hausdorff distance becomes less than screen resolution when i is between 10 and 50.

The attractor attr(C) associated with the code C is, in this way, the image encoded by the code C.

Figure 3 illustrates the decoding process for a four transformation code corresponding to a fern. The initial rectangle which initiates the decoder does not affect the attractor. It could just as well be a sin curve or a random screen. Thirty iterations separate the initial and final images.

The Encoder

The basis for fractal image encoding is the COLLAGE THEOREM: Let B be a target image and let C be an ifs code with contractivity factor $0 < s < 1$. If the Hausdorff distance between B and C(B) is less than E then the Hausdorff distance between B and attr(C) is less than E/(1-s).

This theorem says that to find the ifs compression code for an image or image segment, one can solve the following puzzle. Small (affine) deformed copies of the target must be arranged so that they cover up the target as exactly as possible. This "collage" of deformed copies determines an ifs code since each deformation is an affine transformation of the target. The better the collage, as measured by the Hausdorff distance, the closer will be the attractor of the ifs to the target.

Application of the collage theorem so that E < (1-s)*resolution assures a lossless compression. One can search for transformations which have contractivity factors < .7, for example. Then lossless compression requires E to be less than .3*resolution. More generally, upper bounds on contractivity produces a priori bounds on the errors in the encoded image during the encoding process.

Another consequence of the collage theorem is that if the matrix entries in two codes are close then the attractors of the codes are also close. This has an important interpretation related to error propagation. Small errors in codes lead to small error in images.

Figures 4 and 5 demonstrate the collage theorem.

NASA APPLICATIONS

Data compression can provide service to the NASA mission in both space based and ground based operations.

Data Quality

One issue which arises in space and ground use of lossy compression is that of data quality. Common measures of error in reconstructed data are based on mean-square or root-mean square computations. This type of error calculation is often chosen for convenience, rather than for scientific merit. Scientific analyses attempt to compensate for spatial errors through a registration procedure. Fractal image compression has focused on a different error metric, the Hausdorff distance. It integrates both spatial and spectral data distortion into a single measure of error. Evaluation of the Hausdorff distance as a relevant discriminant of data quality can begin immediately using existing experiment data.

Compression procedures are generally sensitive to transmission bit errors. Sensitivity generally increases with increasing compression ratios. Most high compression ratio techniques are therefore risky in a noisy environment. Fractal image compression contains error propagation independently of the source of the errors. As discussed above the collage theorem provides bounds on the error in the reconstructed data from bounds on errors in the codes. This error containment need not suffer with increased compression ratios.

Scientific Utility

The capture of data from space sensors or experiments and its transmission to earth is a major NASA activity. Often this data is destined to be analyzed by scientists in the form of images. Space sensors and experiments can easily produce enough image data to overload all available data communication bands to earth. Twice as much two-to-one compressed data as uncompressed data can be transmitted in the presence of a bandwidth bottleneck. Fractal compressed images can provide orders of magnitude more images through a bandwidth bottleneck. To achieve 1000-to-1 compression on 1024 x 1024 images, subsampling, for example, would produce 32 x 32 images. Such coarse images may not be of any use to a scientist. Fractal image compression can be used by a scientist to achieve such compression and yet be structured so as to retain certain recognizable features on a fine scale. While such high compression ratio encoding may not be of universal interest, scientists should be given the choice.

High data rate sensors may be operational for only a small percent of their lifetime due to bandwidth bottleneck. Fractal image compression can use additional sensor data during the compression process to increase the quality of the transmitted data.

While cost of storage media decreases and read/write speeds and bandwidth increases, these trends are not able to match the increase in available data. As a consequence, data compression should form an important part of any data management system. Moreover, even given unlimited and inexpensive memory and bandwidth, image analysis would remain as an outstanding problem of overriding importance. Fractal image compression does not simply produce an unintelligible code from uncompressed data. Rather, fractal codes themselves contain geometric and measure-theoretic information about the data sets. Analysis of an image can, in part, be done on the compressed code. In particular, experiments with texture and object identification and classification based on compressed codes can begin with existing experimental data.

Data Management

Interactive access to scientific data increases its usefulness. Users often do not have a specific data address to examine but rather wish to browse through data samples of a generic type. In the browsing mode, only accuracy to some level of detail is required. Fractal image compression can provide interactive browsing on existing networks by simulating a virtual bandwidth orders of magnitude higher than the actual bandwidth.

The proper packaging of data increases its usefulness. Vast amounts of data can be easily interpreted when viewed as animation. High quality animation, however, places an enormous burden on existing and projected communication links. Fractal image compression provides an efficient format for animation. The collage theorem guarantees that animation can be accomplished through small changes in codes that are already highly compressed. Fast decoders will be required to view the animation at video rates. A prototype decoder was displayed in October, 1987 at DARPA's ACMP conference in Washington, D.C. It demonstrated the feasibility of higher performance video rate decoders.

Scientific Justification

The nature of the data collected by space sensors suggest a vast potential for compression, well beyond that indicated by standard entropy calculations. Much of the data is collected from repeated observations over similar areas. Multispectral data is expected to be correlated over multiple channels. The data itself, generated by natural laws, though complex is far from random. As a concluding example, consider the data in Figures 6-9. These graphs of voltage as a function of time measure laser scattering and voltage across a wire in a turbulent jet experiment. Both data streams, which come from the probes of a single system yield the same fractal dimension, 1.5 as indicated in Figure 9. Such correlations are not exploited in classical compression schemes. Fractal image compression can find the hidden redundancy suggested by such data.

$$w_i(x) = w\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} = A_i x + t_i.$$

Then Table 1 is a tidier way of conveying the same iterated function system.

w	a	b	c	d	e	f	p
1	0.5	0	0	0.5	1	1	0.33
2	0.5	0	0	0.5	1	50	0.33
3	0.5	0	0	0.5	25	50	0.34

TABLE 1 IFS code for a Sierpinski triangle.

w	a	b	c	d	e	f	p
1	0.5	0	0	0.5	1	1	0.25
2	0.5	0	0	0.5	50	1	0.25
3	0.5	0	0	0.5	1	50	0.25
4	0.5	0	0	0.5	50	50	0.25

TABLE 2 IFS code for a Square.

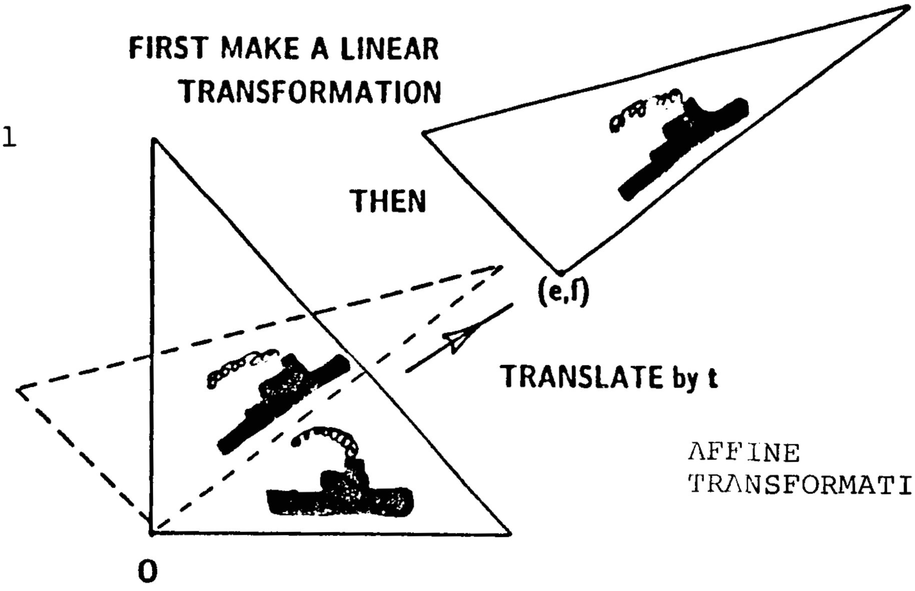
w	a	b	c	d	e	f	p
1	0	0	0	0.16	0	0	0.01
2	0.85	0.04	-0.04	0.85	0	1.6	0.85
3	0.2	-0.26	0.23	0.22	0	1.6	0.07
4	-0.15	0.28	0.26	0.24	0	0.44	0.07

TABLE 3 IFS code for a Fern.

w	a	b	c	d	e	f	p
1	0	0	0	0.5	0	0	0.05
2	0.42	-0.42	0.42	0.42	0	0.2	0.4
3	0.42	0.42	-0.42	0.42	0	0.2	0.4
4	0.1	0	0	0.1	0	0.2	0.15

TABLE 4 IFS code for a Fractal Tree .

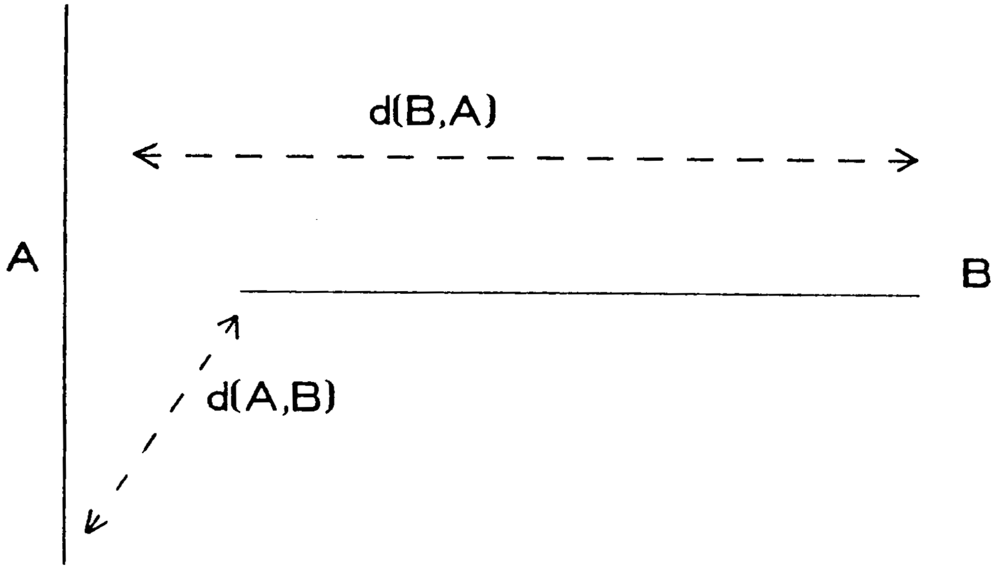
FIGURE 1



AFFINE TRANSFORMATIONS

EXAMPLE:

FIGURE 2



HAUSDORFF DISTANCE is

$$H(A,B) = \text{maximum } \{d(A,B),\ d(B,A)\}$$

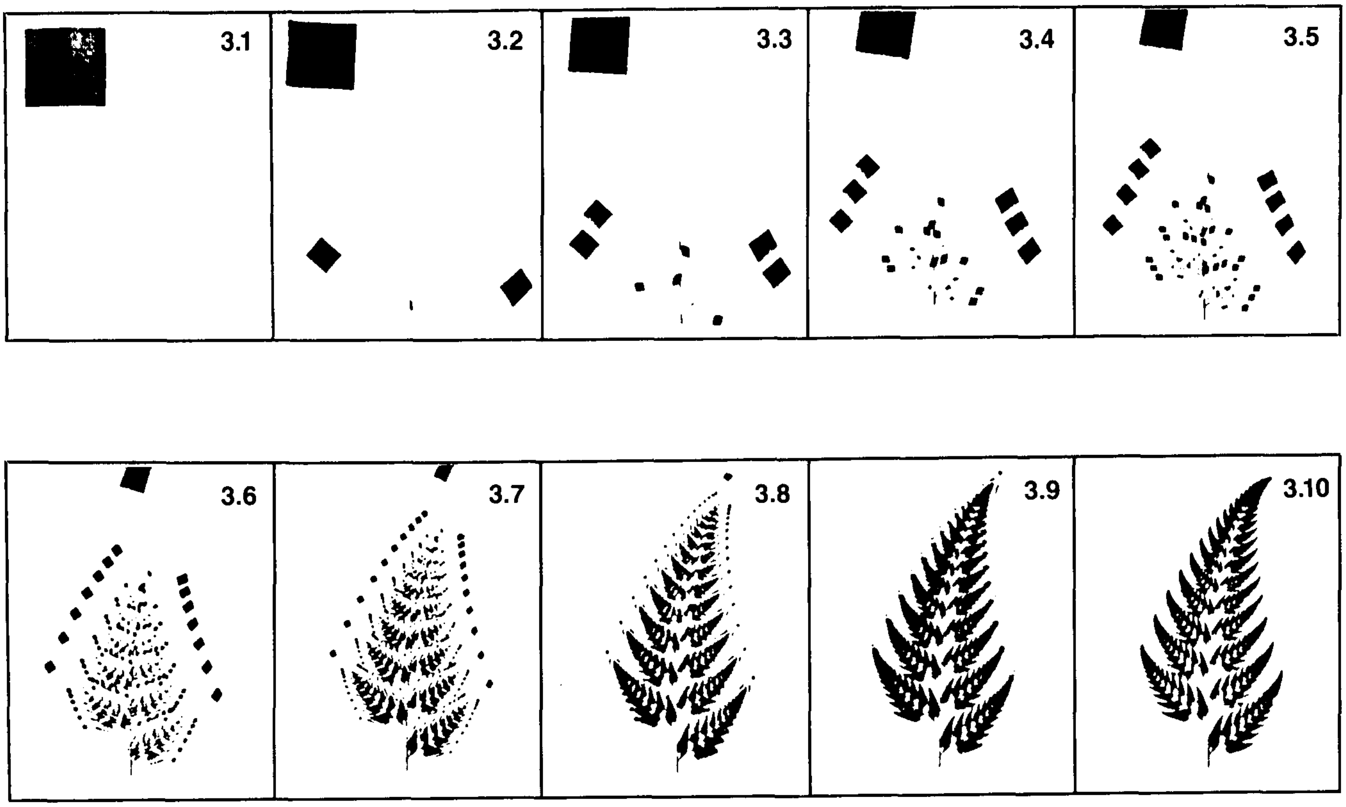
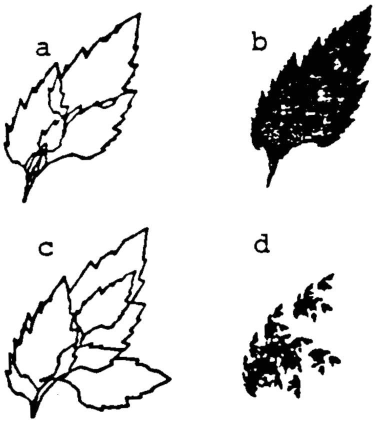


Figure 3

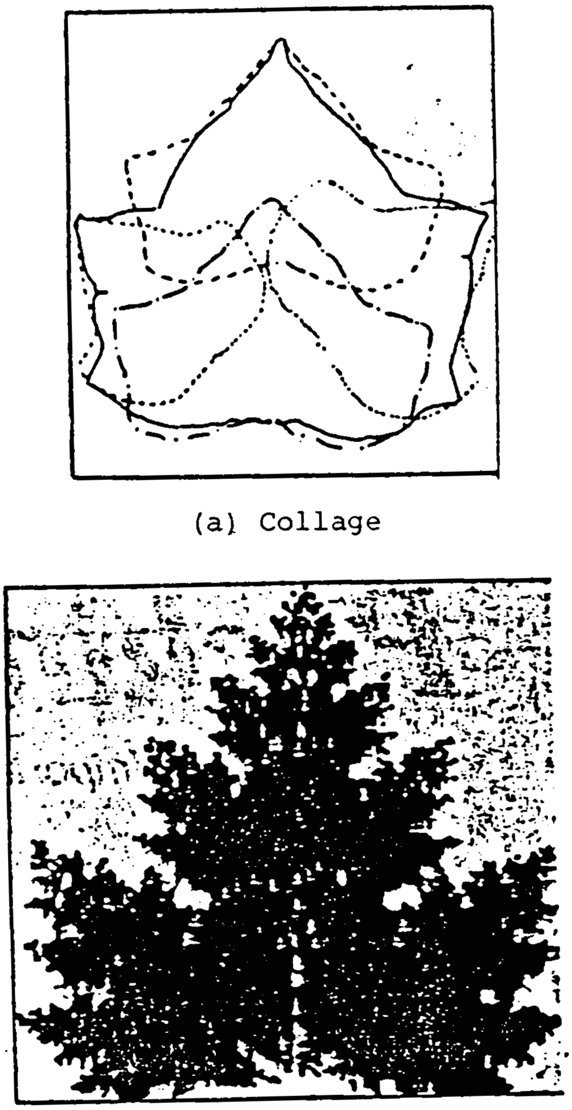
ILLUSTRATION OF DECODING

3.1 is an initial computer screen. This initial state can be chosen at random but in this example it is a small square, in the upper left hand corner of the screen. Four affine transformations are applied to each point in the square and give the four parallelograms in 3.2. A0 is the initial square. The image A1 is the four parallelograms in 3.2. The same four transformations are applied to each of the parallelograms in 3.2 and produce A2 in 3.3 which consists of sixteen parallelograms. Some intermediate screens are not shown. After about thirty iterations the fern appears in Figure 3.10.



Two applications of the collage theorem. (a) and (c) are collages of a leaf under affine transformations. Four transformations are used in each case. The Hausdorff distance between the collage and the target leaf is smaller in (a) than in (c). (b) is the leaf reconstructed from (a) while (d) is reconstructed from (c). The reconstruction from (a) to (b) is superior than that from (c) to (d) as suggested by the collage theorem.

Figure 4



(a) Collage

Collage of four similitudes. Target leaf is outline in solid stroke. Affinely deformed copies have broken outlines.

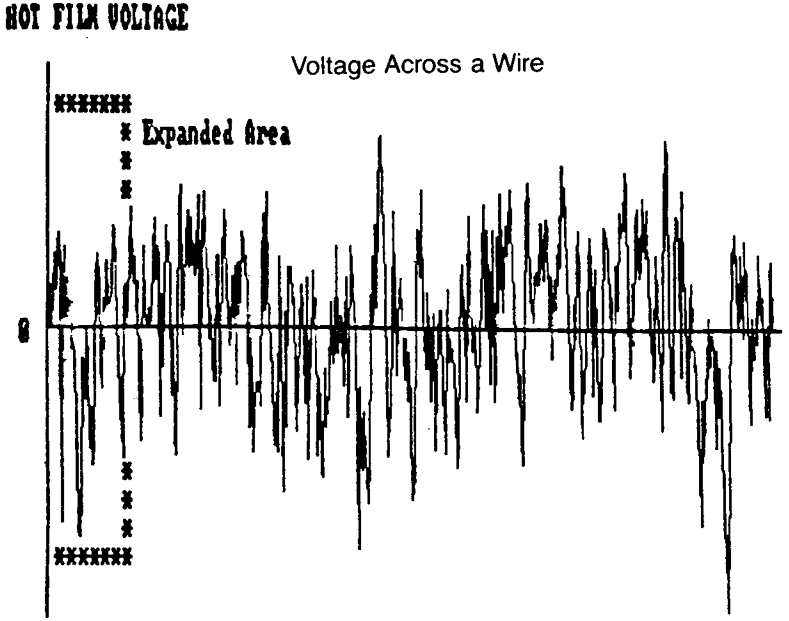
(b) Attractor

Decoded leaf from collage above.

Figure 5

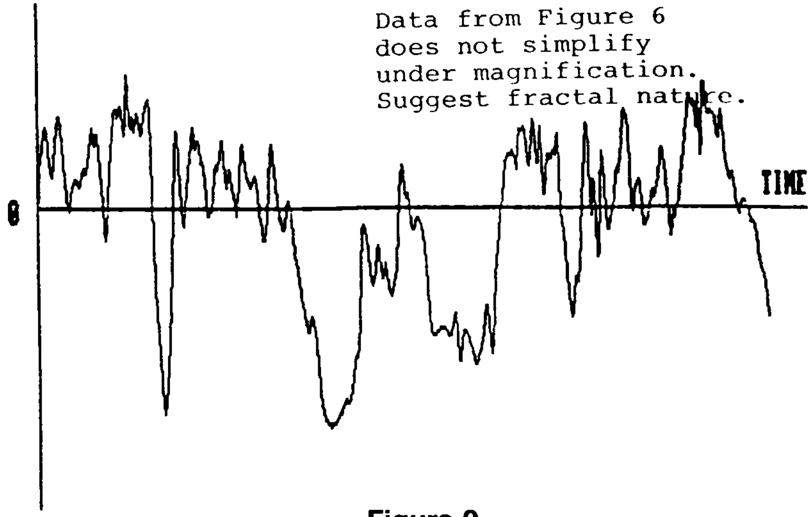
Figures 6-9: Voltage as a function of time in a turbulent jet experiment.

Figure 6



EXPANDED AREA HOT FILM VOLTAGE

Figure 7



Data from Figure 6 does not simplify under magnification. Suggest fractal nature.

RAYLEIGH SCATTERING VOLTAGE

Figure 8

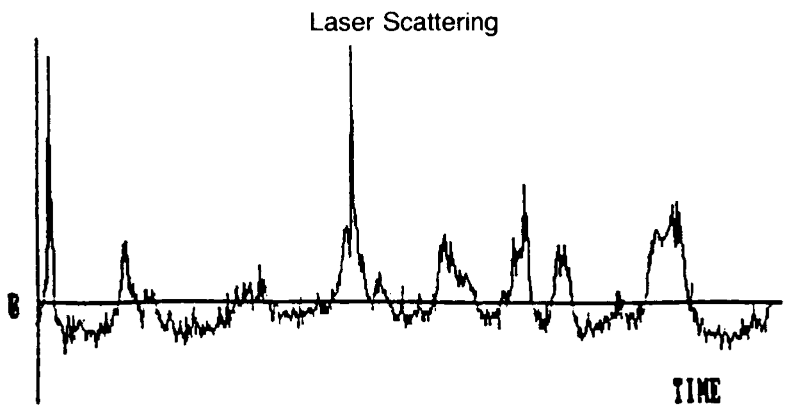
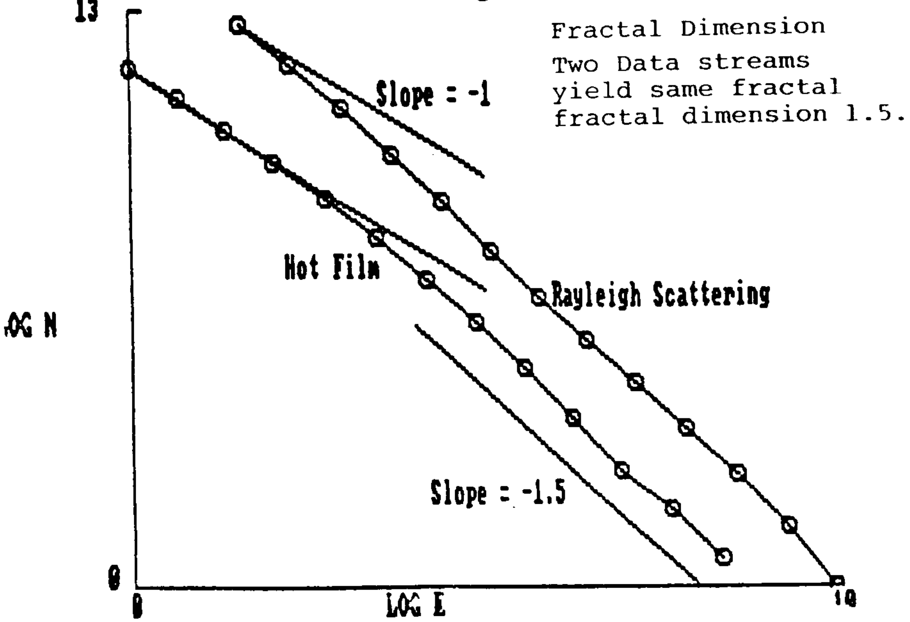


Figure 9



Fractal Dimension

Two Data streams yield same fractal fractal dimension 1.5.

N89-22347

NOISELESS COMPRESSION USING NON-MARKOV MODELS

Anselm Blumer
Computer Science Department
Tufts University

ABSTRACT

Adaptive data compression techniques can be viewed as consisting of a model specified by a database common to the encoder and decoder, an encoding rule and a rule for updating the model to ensure that the encoder and decoder always agree on the interpretation of the next transmission. The techniques which fit this framework range from run-length coding, to adaptive Huffman and arithmetic coding, to the string-matching techniques of Lempel and Ziv. The compression obtained by arithmetic coding is dependent on the generality of the source model. For many sources, an independent-letter model is clearly insufficient. Unfortunately, a straightforward implementation of a Markov model requires an amount of space exponential in the number of letters remembered. The Directed Acyclic Word Graph (DAWG) can be constructed in time and space proportional to the text encoded, and can be used to estimate the probabilities required for arithmetic coding based on an amount of memory which varies naturally depending on the encoded text. The tail of that portion of the text which has been encoded is the longest suffix that has occurred previously. The frequencies of letters following these previous occurrences can be used to estimate the probability distribution of the next letter. Experimental results indicate that compression is often far better than that obtained using independent-letter models, and sometimes also significantly better than other non-independent techniques.

INTRODUCTION

Adaptive data compression techniques are useful when the statistics of the data are not known in advance or are changing slowly. This paper surveys some well-known adaptive compression techniques and presents two new methods which give improved compression in many cases. All techniques presented are lossless (or noiseless) in that an exact copy of the original data can be obtained by decompressing the output of the compression algorithm. The data is assumed to be a string of characters from an arbitrary alphabet, though many of the ideas presented here could be applied to images or other multidimensional data.

An adaptive compression technique can be viewed as consisting of three parts:

1) A dictionary or database which defines the current state of the model.
2) A coding rule which determines how the encoder transmits the next part of the string and how the decoder interprets the encoder's message.
3) An adaptation rule which determines how the encoder and decoder update the model to reflect the previous transmission.

If the encoder and decoder initialize their models identically, they can maintain identical copies of the model throughout the transmission by using the same adaptation rule.

RUN-LENGTH CODING

The simplest compression technique which fits this framework is probably run-length coding. In this case, the dictionary consists of the single character which forms the current run. The coding rule encodes the number of times this character is repeated, followed by an encoding of the next character. For example, if the encoder's and decoder's dictionaries are initialized to "a", the data string "ccccabbb" could be transmitted as (0,c)(3,a)(0,b)(2,b). Since the initial character in the dictionary is "a", (0,c) is interpreted as no "a"s followed by at least one "c". The "c" then replaces the "a" in both dictionaries. (3,a) then indicates a run of 3 more "c"s followed by an "a". (0,b) indicates that this "a" is followed immediately by a "b", and (2,b) can be interpreted to mean that the end of the string has been reached after two more "b"s, since there would not ordinarily be two consecutive runs of the same character.

LEMPEL-ZIV CODING

A more interesting adaptive compression technique was developed by Lempel and Ziv while investigating a complexity measure for strings(4). In this case, the dictionary consists of that portion of the data string which has already been transmitted. The encoder is allowed to specify any substring of this string by transmitting an index and a length. The encoder parses off the longest prefix of the part of the data which has not yet been transmitted which matches a substring of the previously transmitted data. This is complicated by the fact that the encoder and decoder add letters to their dictionary strings as this matching is being done, allowing the matching substrings to overlap each other. The encoder then transmits the index of the previous substring, the length of the match, and the first character which caused the match to fail. If the data string is "abcabbcabbcb" and "abc" has already been transmitted the encoder's next transmission would be (1,2,b) indicating a match of 2 letters starting at position 1, followed by a "b". The encoder and decoder would then concatenate "abb" to their dictionaries, giving "abcabb". Using the overlap mentioned above, the encoder can now transmit the rest of the string as (3,5,b). If the 5 was replaced by a larger number, this would indicate further repeats of the pattern "cabb", so this technique can be viewed in part as a generalization of run-length coding, where runs are allowed to consist of repeated patterns rather than just repeated single characters.

More recently, Ziv and Lempel have developed another adaptive compression technique based on string matching(7). To avoid the complexity inherent in using both a pointer and a length to refer to any previous substring, the dictionaries for this newer method consist of only those substrings which have been parsed and transmitted by the encoder. The prefix parsed by the decoder must match one of these substrings. The decoder then transmits a code number for that substring and the next character. (Variations of this technique avoid the explicit transmission of the next character by methods such as initializing the dictionaries to contain all single-character strings(5). The adaptation rule then adds the most recently transmitted to the dictionaries. For example, if the first eight characters of "aabaaabaaabb" have been transmitted, the dictionaries consist of the strings "a", "ab", "aa", and "aba", numbered 1, 2, 3, and 4, respectively. The encoder will then parse "aa" and transmit "aab" as (3,b). "aab" is then added to the dictionaries as string 5. The final "b" is then transmitted as (0,b), since it does not match any dictionary string. This technique works quite well in practice, as is evidenced by popularity of the UNIX (TM) "compress" command. One slight drawback is that runs of repeated patterns can no longer be parsed all at once, even when the pattern is only the repeated occurrence of a single character.

A GENERALIZATION OF RUN-LENGTH CODING

Another way to avoid transmitting both a pointer and a length is to use an implicit pointer which can be computed by both the encoder and decoder(1). This can be done using a data structure known as the Directed Acyclic Word Graph (DAWG), a data structure which stores information about all substrings of a string in space proportional to the length of the string and can be built in time proportional to this length (linear space and time)(2,3). The linear-time construction algorithm always maintains a pointer to the longest suffix that has occurred elsewhere in the string. This suffix is known as the "tail". The previous occurrence of the tail can be used as the implicit pointer mentioned above. The encoder locates this previous occurrence (which is uniquely defined by the DAWG construction algorithm) and predicts that the next character to be transmitted will be the same as the character which follows this previous occurrence. If this prediction is correct, the DAWG is then updated by adding this character and another prediction is made based on the new tail. This process continues until a prediction fails, at which point the encoder transmits a count of the number of correct predictions and the actual character for the first prediction which failed. For example, if the data string is "aababaabaa" and the first five characters have already been transmitted, then the tail is "ab" and the prediction is that the sixth character is "a". This is correct, so the "a" is concatenated to the string and the DAWG is updated. The tail is now "aba", which predicts a "b". This is incorrect, so the encoder's transmission is (1,a). The next predictions are "b", "a", and "b", of which the first two are correct, so the next transmission is (2,a). With this technique any sufficiently long repeated pattern can be parsed in a single transmission, so it will act like a generalization of run-length coding in these cases.

ARITHMETIC CODING WITH DAWG-BASED MODELS

Although experimental results with this last technique were encouraging, it can be shown to be asymptotically nonoptimal for a wide variety of data sources. Fortunately, the information available from the DAWG can be used in combination with arithmetic coding[6], resulting in a technique which is both asymptotically optimal and gives good results in the cases where the above technique worked well. Arithmetic coding, like Huffman coding, is based on using estimates of the probabilities for the next character to encode that character. An adaptive arithmetic encoder will revise these estimates as each letter is transmitted. If the characters are statistically independent, a table of frequencies for each character will provide these estimates. In most cases, characters are strongly dependent, so a more sophisticated model is needed. The most natural next step is to use a Markov model of order m, where the probabilities of the next character depend on the previous m characters. There are two problems with this: the amount of space needed to store the probability estimates is exponential in m, and the ideal m may change as the characteristics of the data change. Both of these problems can be solved by using a DAWG to store the frequency counts and using the previous occurrence of the tail to provide the appropriate amount of context. Using just the frequency counts from the DAWG gave good compression in many cases, but worked very poorly when the data contained long runs. The reason for this is that in a long run there will be relatively few previous occurrences of the tail, so the next character is not predicted with great certainty. When the probabilities were modified to take into account the length of the tail, the performance improved greatly in these cases.

EXPERIMENTAL RESULTS

Filetype	Filesize	ARITH	LZW	GRL	DAWGARITH
Commands	347	253	200	184	142
C Program	1729	1160	840	603	515
Object Code	3890	2585	2096	2544	1759
Load Module	49152	29.47	23544	27774	18168
Font File	16384	906	478	384	263
This Report	15945	9528	7810	10982	8809

Numbers are number of bytes before and after compression.
ARITH is the adaptive arithmetic coding algorithm from[6].
LZW is the UNIX (TM) "compress" command[5].
GRL is the generalization of run-length coding from[1].
DAWGARITH is the last technique described above.

REFERENCES

1) Blumer, "A Generalization of Run-length Coding," Presented at the IEEE International Symposium on Information Theory, June 1985, Brighton, England.

2) Blumer, Blumer, Ehrenfeucht, Haussler, Chen and Seiferas, "The Smallest Automaton Recognizing the Subwords of a Text," Theoretical Computer Science, (40) 1985, pp. 31-55.

3) Blumer, Blumer, Haussler, McConnell and Ehrenfeucht, "Complete Inverted Files for Efficient Text Retrieval and Analysis," JACM, July 1987, pp. 578-595.

4) Lempel, Abraham and Jacob Ziv, "On The Complexity of Finite Sequences," IEEE Transactions on Information Theory, IT-22, no. 1, Jan. 1976, pp. 75-81.

5) Welch, T.A., "A Technique for High-Performance Data Compression," Computer, 17, no. 6, June 1984, pp. 8-19.

6) Witten, Ian H., Radford M. Neal, and John G. Cleary, "Arithmetic Coding for Data Compression," Communications of the ACM, 30, no. 6, June 1987, pp. 520-540.

7) Ziv, Jacob and Abraham Lempel, "Compression of Individual Sequences via Variable-rate Coding," IEEE Transactions on Information Theory, IT-24, no. 5, Sept. 1978, pp. 530-535.

APPENDIX: THE DAWG CONSTRUCTION ALGORITHM

The DAWG is a Partial Deterministic Finite Automaton (PDFA) which recognizes the set of all substrings of a word. If the word has length n, the DAWG will have less than 2n nodes and 3n edges. It can be built online in linear time using some auxiliary edges (one per node) called suffix pointers. Each node corresponds to the class of strings labeling the paths from the source to that node. The longest such path is called primary. The destination of the suffix pointer of a node is identified by removing the first letter from the label of the shortest path from the source to that node. Thus the suffix pointer of the sink node can be used to identify the longest suffix which occurs somewhere else in the string (the "tail").

The pseudo-C code below updates a DAWG for a word, w to a DAWG for wa. It assumes that wa is stored in a global buffer, and that "index" is the position of the letter a in this buffer. Each node contains:

position: a buffer index pointing to the letter labeling edges to that node
depth : the length of the primary path from the source to that node
edges : a linked list of outgoing edges, and
suffix : the suffix pointer for that node

"source" and "currentsink" are global variables. The following auxiliary procedures are needed:

allocnode(position, depth) allocates and returns a pointer to a new node
allocedge(node, edgelist) adds to "edgelist" an edge which points to "node"
findedge (node, letter) returns a pointer to an outgoing edge from "node" labeled "letter", or NIL if no such edge exists

```
     unlist    ( edgelist, node ) removes an edge pointing to "node"
                 from "edgelist"

update( ch, index )
char ch;               /* ch is the character pointed to by index */
unsigned index;    /* index points into the text buffer       */ {
/* make a new node, "newsink", and make an edge to
/* this node from "currentsink", the old sink.  */
newsink = allocnode( index, index+1 );
edges( currentsink ) = allocedge( newsink, edges( currentsink ) );

suffixnode = source; /* the default value */

/* follow chain of suffix pointers from currentsink */
for (currentnode = suffix( currentsink ); currentnode ISNT NIL;
                  currentnode = suffix( currentnode ) ) {

   /*  no  edge  with  this  character,  so  make  a  secondary  edge  to
"newsink" */    if ((Edge = findedge( currentnode, ch )) IS NIL)
      edges( currentnode ) =
         allocedge( newsink, edges( currentnode ) );

   /* a secondary edge labelled "ch", so split */
   else if ((depth(currentnode)+1) ISNT depth(node(Edge))) {

      /* Make Edge into a primary edge to a new node */
      childnode = node ( Edge );
      newchildnode = allocnode( index, depth(currentnode)+1 );
      edges( currentnode ) = unlist( edges( currentnode ), childnode
);       edges( currentnode ) =
         allocedge(  newchildnode, edges( currentnode ) );

      /* Give a copy of each of childnode's edges to newchildnode */
    for (Edge = edges( childnode ); Edge ISNT NIL; Edge = next( Edge
))          edges( newchildnode ) =
            allocedge( node( Edge ), edges( newchildnode ) );
      /* Set the suffix pointer of newchildnode to that of childnode,
     /* and reset childnode's to point to newchildnode     */
      suffix( newchildnode ) = suffix( childnode );
      suffix( childnode ) = newchildnode;

      /* follow chain of suffix pointers, changing secondary edges
  /* which point to childnode to point to newchildnode      */
for (currentnode = suffix( currentnode ); currentnode ISNT NIL;
     currentnode = suffix( currentnode )) {
         if ((Edge = findedge( currentnode, ch )) IS NIL) break;
   if ((depth(currentnode)+1) IS depth(node(Edge))) break;
edges( currentnode ) = unlist( edges( currentnode ), node( Edge ) );
       edges( currentnode ) =
            allocedge( newchildnode, edges( currentnode ) );
      }
      suffixnode = newchildnode;
      break;
   }
   /* otherwise, it's primary, so set suffixnode and break out */
else {
      suffixnode = node( Edge );
      break;
   }
}
/* set the suffix of newsink to be the node the above "for" loop found
*/ suffix( newsink ) = suffixnode;
currentsink = newsink;
}
*7*
```

PERFORMANCE OF LEMPEL-ZIV COMPRESSORS WITH DEFERRED INNOVATION

Martin Cohn
Computer Science Department
Brandeis University

INTRODUCTION

The noiseless data-compression algorithms introduced by Lempel and Ziv(6,7) parse an input data string into successive substrings each consisting of two parts: The citation, which is the longest prefix that has appeared earlier in the input, and the innovation, which is the symbol immediately following the citation. In "extremal" versions of the LZ algorithm the citation may have begun anywhere in the input; in "incremental" versions it must have begun at a previous parse position. Originally the citation and the innovation were encoded, either individually or jointly, into an output word to be transmitted or stored. Subsequently, it was been speculated by several authors(2,4,5,7) that the cost of this encoding may be excessively high because the innovation contributes roughly lg(A) bits, where A is the size of the input alphabet, regardless of the compressibility of the source. To remedy this excess, they suggested storing the parsed substring as usual, but encoding for output only the citation, leaving the innovation to be encoded as the first symbol of the next substring. Being thus included in the next substring, the innovation can participate in whatever compression that substring enjoys. We call this strategy deferred innovation. It is exemplified in the algorithm described by Welch(5) and implemented in the C program compress that has widely displaced adaptive Huffman coding (compact) as a UNIX system utility.

While compress achieves respectable compression ratios on highly compressible data (say two-to-one or better), it performs poorly, compared to theory and to other versions of LZ compression, on relatively incompressible data. In the extreme of total incompressibility, such as uniform i.i.d. or well encrypted data, compress frequently expands the input by about 45% when the output word size is 12 bits and by about 90% when the output word size is 16, to mention two common options.[2]

These figures stand in contrast to LZ realizations without deferred innovation, where random data are expanded by about 5% for output words of 12 or more bits. The purpose of this paper is to explain the excessive expansion, and implicitly to warn against the use of deferred-innovation compressors on nearly incompressible data.

Suppose a deferred-innovation LZ algorithm operates on a string of b-bit input characters producing B-bit output words,[3] and assume that as in most implementations the dictionary of citations is initialized with all the individual symbols of the input alphabet. For an input string "x y x z . . ." such an algorithm will output B bits for the first "x", and store "xy"; then output B bits for "y", and store "yx"; then output B bits for "x", and store "xz"; and so on. In general, B bits will be output for every position that initiates a novel pair, that is, a pair not seen earlier. What we shall show in the next part of the paper is that if the input length is much less than the square of the alphabet size, the "typical" string of length N has almost N novel pairs, and therefore the output length must be almost NB, and the compression ratio almost B/b. Now, when the input is a string over the alphabet of 256 bytes, the input length would have to be comparable to 2^{16} to avoid this condition; otherwise the compression ratio will likely be close to 12/8 = 1.5 or 16/8 = 2.0 for common choices of B. This is just the behavior mentioned above for the program compress. The "typical" string is generated by a uniform

[2] In default mode, compress refuses to recode a file doomed to expansion.

[3] While ideally this output word would be variable in length, it is easy to show that not much is gained by the complication, so we shall conform to practice and make the wordlength fixed.

independent source over the alphabet, or selected uniformly from among all the possible strings of length N.

We say an ordered pair of consecutive input symbols is novel if and only if the identical ordered pair has not appeared earlier in the input string. Let $S_i(A,N)$ be the number of strings of length N over an alphabet of size A, with a novel pair beginning at position i. The pair beginning in position i can be repeating, like "xx", or nonrepeating, like "xy". Thus $S_i(A,N)$ consists of two parts, according to whether the novel pair at position i is repeating or not; see Figure 1.

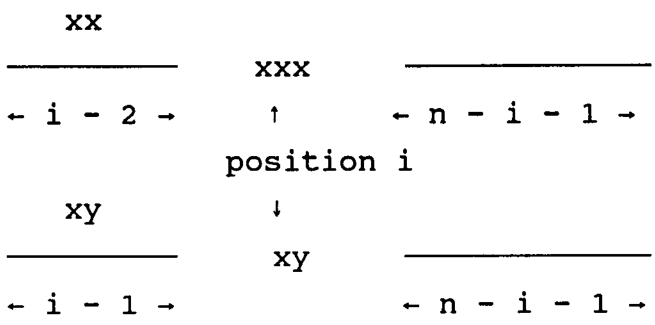


FIGURE 1

Clearly $S_1(A,N) = A^N$, and

$$S_i(A,N) = (A - 1)A^{N-i-1}D(A,i - 2) + (A - 1)A^{N-i}E(A,i - 1),$$

where D(A,i-2) counts strings of length i-2 containing no "xx", and E(A,i-1) counts strings of length i-1 containing no "xy", while x and y range over the alphabet. (We assume the indices are nonnegative and take D(A,0) = E(A,0) = 1 by convention).

Figure 2 shows, for the respective processes (or languages) that contain no repeating pair "xx" or no nonrepeating pair "xy", the state diagrams, adjacency matrices, characteristic equations, and the latters' roots.

$$\begin{bmatrix} A - 1 & 1 \\ A - 1 & 0 \end{bmatrix} \qquad\qquad \begin{bmatrix} A - 1 & 1 \\ A - 2 & 1 \end{bmatrix}$$

$$\delta^2 - (A - 1)\delta - (A - 1) \qquad\qquad \epsilon^2 - A\epsilon + 1$$

$$\delta = \frac{A - 1 \pm \sqrt{(A + 1)^2 - 4}}{2} \qquad\qquad \epsilon = \frac{A \pm \sqrt{A^2 - 4}}{2}$$

FIGURE 2

From linear theory,

$$D(A,k) = d_+\delta_+^k + d_-\delta_-^k \text{ and } E(A,k) = e_+\epsilon_+^k + e_-\epsilon_-^k,$$

where d_+, d_-, e_+, and e_- are constants determined by the initial conditions:

$$D_0(A) = E_0(A) = 1, \; D_1(A) = E_1(A) = A.$$

In particular

$$\epsilon_+ = A - A^{-1} - \Theta(A^{-3}), \; \epsilon_- = A^{-1} + \Theta(A^{-3}),$$
$$e_+ = 1 - \Theta(A^{-2}), \text{ and } e_- = \Theta(A^{-2}).$$

We can now estimate $S(A,N)$, the total number of novel pairs among all strings of length N, and $A^{-N}S(A,N)$, the average number of novel pairs per string. We underestimate $S(A,N)$ by ignoring $D(A,N)$, which is positive. Likewise, since e_- and ϵ_-^k are positive, we underestimate by ignoring them. This leaves the approximation

$$S(A,N) \geq A^N + (A -1)A^{N-1} \sum_{i-2}^{N} A^{-i+1}E_{i-1}$$

$$= A^N + (A-1) \sum_{j=1}^{N-1} e_+\epsilon_+^j A^{N-j-1}$$

$$\geq A^N = (A - 1)e_+\epsilon_+ [A^{N-2} + A^{N-3} \epsilon + \ldots + \epsilon^{N-2}]$$

Multiplying and dividing by N-1, and using the fact that the arithmetic mean dominates the geometric mean, we have

$$\begin{aligned} S(A,N) &\geq A^N + (A - 1)e_+\epsilon_+(N -1)(A\epsilon_+)^{(N-2)/2} \\ &=(1 - \Theta(A^{-1}))\ [A^N + (N -1)A^N\ (1 -1/A^2)^{(N-2)/2}] \end{aligned}$$

In the limit of large N this last expression is well approximated by

$$A^N + (N -1)A^N \exp\ (-N + 2/2A^2).$$

Division by A^N confirms the claim that the average number of novel pairs per string of length N remains about N until the string length exceeds the square of the alphabet size.

A simpler but similar calculation can be used to estimate the expected number of novel singletons (symbols) in a string of length N. As before, let $S_i(A,N)$ be the number of sequences in which the ith symbol is novel. Let that symbol be "x"; then the previous i-1 symbols may be anything but x, and the succeeding N-i symbols may be anything at all. Thus

$$S_i(A,N) = (A - 1)^{i-1}A^{N-i+1} \text{ and } S(A,N) = \sum_{i=1}^{N} S_i(A,N).$$

As before, this can be underestimated by the geometric mean to give:

$$S(A,N) \geq AN\ (\ A^{(N-1)/2}(A-1)^{(N-1)/2})$$

which is asymptotically

$$NA^{N} \exp -(N - 1/2A).$$

Thus the average number of novel singletons (symbols) in a sequence of length N remains about N until N exceeds the alphabet size. Similar arguments may be used as well to show that almost all k-tuples will be novel until the sequence length exceeds the k^{th} power of the alphabet size.

DISTRIBUTION OF MEMORY CONTENTS

We next consider the distribution of pairs, triples, and higher-order tuples in the L/Z compressor memory during three regimes: While the memory is filling but not yet full; when it has just filled; and when it is full and in equilibrium. Our assumption is still that input symbols are selected uniformly and independently over some finite alphabet. Another assumption must be made, regarding possible deletions from the memory once it has filled. In practice a variety of deletion strategies have been used, notably l.r.u., whereby the least-recently used entry is deleted to make room for the newest insertion. In this paper we will usually make the simpler assumption that the entry to be deleted is chosen randomly from among the non-singletons. In other words, deletion is random except that the alphabetic symbols are immune.

Memory Filling

Initially the compressor memory (or dictionary) contains α singleton entries, namely the symbols themselves. Each time a match to a singleton is found a pair is inserted; should a pair be matched, its extension to a triple is inserted, and so on. Given the uniform, independent input assumption, it is clear that the likelihood of matching a given pair is only $1/\alpha$ times the likelihood of matching a singleton. Since we are interested mainly in large values of α like 32, 64, 128, 256, we will ignore the possibility of creating quadruples or higher-order tuples, and lump them with the triples. Thus the memory at any time contains α singles, β pairs, and γ others. Let λ be the total number of memory locations, and let $\mu = \lambda$ - alpha be the number of locations available for pairs and higher-order tuples. Then at the time of the t^{th} insertion we have

$$\beta + \gamma = t \text{ for } t < M, \quad \beta + \gamma = \mu \text{ for } t \geq \mu.$$

The distribution of β (hence γ) at time t during filling is given by the formula

$$\Pr\{\beta|t\} = \alpha^{-t} S(t,\beta)\alpha_{(\beta)}$$

where $S(t,\beta)$ is a Stirling number of the second kind, and $\alpha_{(\beta)}$ is a falling factorial of α^3. The reason is that there are $S_{t,\beta}$ ways of choosing a sequence of t symbols which includes exactly β distinct symbols, and that the identities of those β distinct symbols can be chosen in exactly $\alpha_{(\beta)}$ ways. This count is then divided by α^t, the total number of ways of choosing a sequence of t symbols from the alphabet. Since μ is a large number for any reasonable compressor, we really need the asymptotic distribution in order to analyze the possibly transient behavior when the memory has just filled, but we don't know it at this time.

Transient Period Under L.R.U. Deletion

When the memory has just filled with pairs and higher-order tuples we speculate that there might be interaction between insertion and deletion by the l.r.u. rule that could cause temporary instability.

In particular, if the memory size is close to α^2 then most of the earliest arrivals will have been pairs, and many of the recent arrivals will be triples, as a result of pairs having been matched and extended. This suggests a "gradient" from most recent to least recent shading from triples to pairs. In such a case, under l.r.u.-deletion, disproportionately more pairs will be deleted, abnormally increasing the proportion of non-pairs until the inability to match triples causes pairs to be recreated and reinserted. The alternation in proportions of pairs versus higher-order tuples would likely damp out. This transient behavior has not been confirmed, but is a topic of ongoing research. The distinction between this hypothesis and the equilibrium analyses below stems from the l.r.u. deletion policy, which makes critical not just the distribution of pairs and non-pairs, but their arrangement in memory as well.

Equilibrium State and Distribution

Finally we consider the distribution of memory contents and the compression ratio at equilibrium. We once again invoke the assumption of random deletion (contrary to the l.r.u. rule used in the previous, speculative section). First we solve for an equilibrium state, that is, a ratio of pairs to non-pairs that is stable, and then we generalize to an equilibrium distribution of probabilities of ratios.

Equilibrium State

Suppose that the memory is full, that it contains β pairs (and thus μ - β non-pairs) and that a randomly chosen input pair is read. The probability that the input matches some pair in the memory is β/α^2. The probability that some pair (rather than a triple) is chosen for deletion is β/μ. Since these are independent events, the four joint probabilities for the change in β are:

$$\Delta\beta = \left[\begin{array}{ll} (+1) \cdot \frac{\alpha^2 - \beta}{\alpha^2} \cdot \frac{\mu - \beta}{\mu} & \text{gain a pair, lose a triple} \\ 0 \cdot \frac{\alpha^2 - \beta}{\alpha^2} \cdot \frac{\beta}{\mu} & \text{gain a pair, lose a pair} \\ 0 \cdot \frac{\beta}{\alpha^2} \cdot \frac{\mu - \beta}{\mu} & \text{gain a triple, lose a triple} \\ (-1) \cdot \frac{\beta}{\alpha^2} \cdot \frac{\beta}{\mu} & \text{gain a triple, lose a pair} \end{array}\right.$$

At equilibrium the first and last probabilities must be equal, and we can solve for β: (Recall that we are ignoring the creation of quadruples or high-orders).

$$\beta = \frac{\alpha^2\mu}{\alpha^2 + \mu}; \qquad \gamma = \mu - \alpha = \frac{\mu^2}{\alpha^2 + \mu}$$

Using these values we can estimate the compression ratio achieved in this equilibrium state by compress, which will output B bits (12 by default) for each 2b bits in a pair can be matched, and will output B bits for only b bits in when a pair cannot be matched. The ratio at equilibrium is thus

$$\rho_{eq} = \frac{(\alpha^2 - \beta)B + \beta B}{(\alpha^2 - \beta)b + 2\beta B} = \frac{\alpha^2 B}{(\alpha^2 + \beta)b} = \frac{(\alpha^2 + \mu)\lg\mu}{(\alpha^2 + 2\mu)\lg\alpha} + O(1/\alpha).$$

From this expression we would expect compress in default mode (with b=8, B=12) to have $\rho_{eq} = 1.42$, which is quite close to experience.

Equilibrium Distribution

We consider next the equilibrium distribution governing the number of pairs present in the compressor memory. Again we assume that the probabilities of creating quadruples, quintuples, and so on are negligible, so that they can be lumped together with the triples. As usual, the singletons are permanent memory residents.

With memory size μ consider the random variable β describing the number of pairs present. β ranges from 0, when the memory has no pairs, to min (μ,α^2) when either the memory is full of pairs, or all pairs are present. As each parse of the uniform, independent input is made, β may increase by 1 or decrease by 1 (except at the extremes) or stay the same, with the respective probabilities given in Section C1. above[4]. This gives us a Markov process with transition matrix

$$T_{i,j} = \begin{cases} \dfrac{i^2}{\alpha^2\mu} & j = i - 1, \\[2ex] \dfrac{i^2}{\alpha^2} + \dfrac{i}{\mu} + -2\,\dfrac{i^2}{\alpha^2\mu} & j = i, \\[2ex] (1 - \dfrac{i}{\alpha^2})(1 - \dfrac{i}{\mu}) & j = i + 1, \\[2ex] 0 & \text{otherwise.} \end{cases}$$

Because this is a connected Markov process, it has an equilibrium distribution p[1] which satisfies $pT = p$, or $(pT)_i = p_i$. We show in the Appendix that

[4] This distribution was erroneously described in the Snowbird talk as an Ehrenfest model[1]. As we shall see, it is rather like a componentwise product of two Ehrenfest processes.

$$p_i = \genfrac{}{}{0pt}{}{\alpha^2}{i} \genfrac{}{}{0pt}{}{\mu}{i} \Big/ \sum_{0}^{\min(\alpha^2,\mu)} \genfrac{}{}{0pt}{}{\alpha^2}{i} \genfrac{}{}{0pt}{}{\mu}{i}$$

For a very simple example, let $\alpha^2 = 4$, $\mu = 5$. Then

$$T = \begin{bmatrix} 0 & 20 & 0 & 0 & 0 \\ 1 & 8 & 12 & 0 & 0 \\ 0 & 4 & 10 & 6 & 0 \\ 0 & 0 & 9 & 9 & 2 \\ 0 & 0 & 0 & 16 & 4 \end{bmatrix}$$

$$p(i) = \frac{1}{16}(1,4,6,4,1) \text{ x } \frac{1}{31}(1,5,10,10,5) = \frac{1}{126}(1,20,60,40,5).$$

This means that asymptotically the distribution is the product of two Gaussian distributions, with relatively displaced means unless $\alpha^2 = \mu$.

ACKNOWLEDGEMENT

I am grateful for conversations on this work with Ira Gessel and Jim Storer. Dana Goldblatt conjectured the dictionary behavior that led to the study of pair/nonpair distributions.

APPENDIX

It suffices to deal with $\mathbf{q} = \mathbf{p}\ \Sigma\ p(i)$ and to show that

$$(qT)_i = \mathbf{q}_{i-1}T_{i-1,i} + \mathbf{q}_iT_{i,i} + \mathbf{q}_{i+1}T_{i+1,i} = \genfrac{}{}{0pt}{}{\alpha^2}{i}\genfrac{}{}{0pt}{}{\mu}{i} = \mathbf{q}_i.$$

$$(qT)_i = \genfrac{}{}{0pt}{}{\alpha^2}{i-1}\genfrac{}{}{0pt}{}{\mu}{i-1} \quad 1 - \frac{i-1}{\alpha^2} \quad 1 - \frac{i-1}{\mu}$$

$$+ \genfrac{}{}{0pt}{}{\alpha^2}{i}\genfrac{}{}{0pt}{}{\mu}{i}\left[\frac{i}{\alpha^2} + \frac{i}{\mu} - \frac{2i^2}{\alpha^2\mu}\right] + \genfrac{}{}{0pt}{}{\alpha^2}{i+1}\genfrac{}{}{0pt}{}{\mu}{i+1}\frac{(i+1)^2}{\alpha^2\mu}$$

$$= \frac{1}{\alpha^2\mu}\left[\genfrac{}{}{0pt}{}{\alpha^2}{i-1}\genfrac{}{}{0pt}{}{\mu}{i-1}(\alpha^2 - i + 1)(\mu - i + 1)\right.$$

$$\left. + \genfrac{}{}{0pt}{}{\alpha^2}{i}\genfrac{}{}{0pt}{}{\mu}{i}(\alpha^2 i + \mu i - 2i^2) + \genfrac{}{}{0pt}{}{\alpha^2}{i+1}\genfrac{}{}{0pt}{}{\mu}{i+1}(i+1)^2\right]$$

$$= \frac{1}{\alpha^2\mu}\genfrac{}{}{0pt}{}{\alpha^2}{i}\genfrac{}{}{0pt}{}{\mu}{i}[i^2 + \alpha^2 i + \mu i - 2i^2 + (\alpha^2 - i)(\mu - i)]$$

$$= \genfrac{}{}{0pt}{}{\alpha^2}{i}\genfrac{}{}{0pt}{}{\mu}{i}.$$

Q.E.D.

C-5

REFERENCES

1) Feller, William, "An Introduction to Probability Theory and its Applications", John Wiley & Sons Inc., New York, Vol.1, 1950.

2) Miller, V.S. and M.H. Wegman, "Variations on a Theme by Lempel and Ziv", Combinatorial Algorithms on Words, Springer-Verlag (A. Apostolico and Z. Galil, editors), pp. 131-140, 1985.

3) Riordan, John, "An Introduction to Combinatorial Analysis", John Wiley & Sons, Inc., New York, 1958.

4) Storer, J.A. and T.G. Szymanski, "Data Compression Via Textual Substitution", J.ACM 29, 4, pp. 928-951, 1982.

5) Welch, T.A., "A Technique for High-Performance Data Compression", IEEE Computer 17, 6, pp. 8-19, 1984.

6) Ziv, J. and A. Lempel, "A Universal Algorithm for Sequential Data Compression", IEEE Trans. Inf. Theory IT-23, 3, pp. 337-343, 1977.

7) Ziv, J. and A. Lempel, "Compression of Individual Sequences Via Variable Rate Coding", IEEE Trans. Inf. Theory IT-24, 5, pp. 530-536, 1978.

DATA COMPRESSION HARDWARE

DAVE EISENMAN

JET PROPULSION LABORATORY

SESSION COORDINATOR

ALASKA SAR FACILITY (ASF)
SAR COMMUNICATIONS (SARCOM)
DATA COMPRESSION SYSTEM

Stephen A. Mango
Digital Image Processing Laboratory
Naval Research Laboratory

INTRODUCTION

The basic technical goal of the Alaska SAR Facility SAR Communications system (ASF SARCOM) is to provide a real-time operational, applications demonstration of the transmission of spaceborne synthetic aperture radar (SAR) imagery of Arctic ice over a bandwidth-limited communications satellite link.

The imagery is to be transmitted from the ASF located at the University of Alaska in Fairbanks to the National Oceanic and Atmosphere Administration (NOAA) Ice Center (NIC) in Suitland, Maryland via the DOMSAT link.

The SARCOM system will be designed to handle the spaceborne SAR imagery of the three following non-U.S., polar orbiting platform/sensors:

1. E-ERS-1 (European Space Agency) April, 1990 Launch
2. J-ERS-1 (Japan) 1992 Launch
3. RADARSAT (Canada) 1994 Launch

The SARCOM system will be able to handle the spaceborne SAR imagery in all high and low resolution modes of the three SAR systems over their operational lifetimes.

The need for data compression in the SARCOM system is driven by two factors:

a) the need to transmit in a real-time operation the high resolution imagery of the SAR sensors with their high data rates of 40-60 megabits per second (Mbps), and

b) the constraint imposed by the bandwidth limitation of the communications satellite link, namely 1.33 Mbps maximum.

These factors imply the need for data compression techniques with effective compression ratios as high as 30-to-1 and 45-to-1, respectively. All techniques known to produce such high compression ratios for any reasonable imagery data have been traditionally categorized as irreversible or fidelity (information) reducing techniques. Furthermore, many of these techniques impose a very high arithmetic load on the real-time system used to implement the data compression coding.

This article will describe how the real-time operational requirements for SARCOM translate into a high speed image data handler and processor to achieve the desired compression ratios and the selection of a suitable image data compression technique with as low as possible fidelity (information) losses and which can be implemented in an algorithm placing a relatively low arithmetic load on the system.

OVERVIEW OF THE SARCOM SYSTEM

Figure 1 is a pictorial of the SARCOM data handling scenario. A generic, spaceborne SAR is portrayed, representing either of the three sensors, viewing a portion of the Arctic basin. It is shown operating in the customary strip map mode indicated by the dashed lines. These dashed lines represent the edges of the data collection swathwidth as the SAR footprint looks broadside to one side of the subsatellite track.

All three of the SAR platforms will be in a polar orbit providing excellent opportunities to provide synoptic coverage of the Arctic basin and land regions. Overlaps in the operating lifetimes of the three systems should provide unique, spaceborne observations of these regions at multiple frequencies, polarizations and look angles. Table 1 presents the essential descriptions of the E-ERS-1, J-ERS-1, and RADARSAT including the basic SAR characteristics, the orbit and the mission(1). In this presentation the notation high resolution or HI-RES will be used to denote a nominal ground resolution of 30 meters by 30 meters for a four-look azimuth processing, one-look range processing, while low resolution, or LO-RES, will denote a nominal ground resolution of 240 meters by 240 meters derived from an 8-by-8 averaging of the HI-RES data. The pixel spacing will be 12.5 meters square and 100 meters square for the HI- and LO-RES imagery, respectively.

Figure 1 indicates the SAR raw data collected onboard being downlinked to the ASF at the University of Alaska at Fairbanks (UAF). These raw data received at the ASF are basically, accurately time-tagged radar backscattered, amplitude returns or the so-called radar echoes from successive pulses. If these radar echoes are considered in a form of amplitude as a function of the two SAR spatial dimensions -- slant range (crosstrack) and azimuth (alongtrack) -- they represent essentially a two-dimensional amplitude interferogram which is analogous to the two-dimensional intensity hologram used in three-dimensional, coherent laser holography.

The main system at the ASF will be used to perform the technically demanding, numerically intensive two-dimensional, coherent SAR processing to convert the raw data echoes into the more familiar two-dimensional SAR images. This conversion from the raw data domain to the image domain requires not only the time-tagged radar echoes but also the radar operating parameters, the SAR platform orbit and attitude data and the earth's geoid size, shape, and motion data. The ASF will perform this SAR processing in a delayed start, near real-time mode. At the present time the start delay reflects primarily the time required to receive and input a sufficiently accurate ephemeris for the SAR platform orbit.

Figure 2 portrays the functional block diagram for the basic ASF system comprised of three major systems: the Receiving Ground System (RGS), a two part SAR Processor System (SPS), and a two part Archive and Operations System (AOS). The main system at the ASF is being funded by the National Aeronautics and Space Administration (NASA). The ASF physical site at the Geophysical Institute is being funded by the University of Alaska at Fairbanks (UAF). Jet Propulsion Laboratory (JPL) is the designer and prime contractor for providing the end-to-end main ASF system. The ASF will be operated by the UAF.

SARCOM will be a stand-alone image processing and image handling system that will be collocated with the ASF system. It will be an attachment and augmentation to the main ASF system for the primary purpose of accessing the quick turnaround, low and high resolution ice imagery processed by the ASF system to be delivered to the NOAA Ice Center. SARCOM will acquire and process these image data at real-time rates.

The end-to-end SARCOM system is depicted in the pictorial labeled Figure 3. The data "umbilical cords" to the main ASF are also displayed.

From the ASF site SARCOM will link the SAR data via a T1 microwave link to the NOAA Tracking and Data Acquisition (TDA) station at Gilmore Creek at real-time rates as shown in Figure 1. Gilmore Creek is located approximately 12-14 miles northeast of the UAF site. For the SARCOM operation the usual 1.54 Mbps T1 link bandwidth limitation will be constrained to an overall maximum transfer rate of 1.33 Mbps.

From the NOAA TDA station the SAR data will be skylifted to the domestic satellite system (DOMSAT) via another T1 microwave link and relayed in real-time to the NIC in Suitland, Maryland. The data will actually be ingested by the existinq NOAA Command and Data Acquisition

(CDA) station and passed to the Ice Center under the auspices of the National Meteorological Center which are all collocated in the same building complex in Suitland.
The LO-RES data can be displayed and/or post-processed for information extraction and analysis immediately upon reception at the NIC for the original LO-RES imagery didn't need to be compressed prior to the transmission in real-time using the SARCOM 1.33 Mbps link. The 64-to-1 reduction of the 8-by-8 averaging of the HI-RES data to derive the LO-RES data corresponds to at most 0.94 Mbps for the LO-RES for even the most demanding 60 Mbps maximum HI-RES mode; thus the LO-RES data will always fit within the link bandwidth constraint. However, the HI-RES image data will have to be compressed/coded using a data compression algorithm at the SARCOM transmit end in Alaska to fit within the link bandwidth and then the image must be reconstructed by using the inverse algorithm at the NIC receive end in Suitland.

The SARCOM operational applications goal is to provide SAR ice imagery to enhance and expand the ice forecasting capabilities and services of the Ice Center. It is planned that this imagery will serve as both quick forecasting and synoptic data bases for direct ship support operating in these areas, longer term sea ice modelling and applied research areas. These data will be utilized in the determination of ice concentration, classification of ice types and the determination of ice motion for an understanding of the kinematics of ice fields and the incorporation into ice dynamics modelling.

FUNCTIONAL REQUIREMENTS OF THE SARCOM SYSTEM

The basic functional requirements for the SARCOM system at the ASF are indicated in Figure 4. The crucial input data for the SARCOM application will be the LO-RES and HI-RES Quick-Turnaround Data which will both be stored on Ampex DCRSi (Digital Cassette Recording System) cassettes.

The Quick-Turnaround Imagery is imagery processed by the main ASF

system that is less than six hours old from the time of the actual capture of the raw data echoes used to form the images. These "fresh" images are a requirement in order that the Ice Center can utilize them in their quick forecasting for direct ship traffic support.
The SARCOM maximum real-time input rate that must be sustainable over the system transfer time for the scene is 8 megabytes per second with 1 byte per pixel, up to 12,000 pixels per swathline and up to 640 swathlines per second.

Therefore, the real-time SARCOM system must have an effective throughput rate of up to 8 megapixels per second. Using the results of a data compression study conducted at the Digital Image Processing Laboratory, the algorithm(s) specified for the compression of the HI-RES data, the most demanding mode of operation for the SARCOM, will have to support an arithmetic load in the range of 20-25 real operations per input pixel or an effective arithmetic rate of 160-200 million floating point operations per second (MFLOPS).

The compressed representation of the HI-RES images and the uncompressed LO-RES images must be outputted from SARCOM in the DOMSAT/NOAA data ingest center formats at a maximum rate of 1.33 Mbps (that is 167 kilobytes per second maximum). There will be 1 byte per compressed coefficient for the HI-RES data and 1 byte per uncompressed image pixel for the LO-RES data.

The SARCOM functional components previously illustrated in Figure 3 consist of:

1. a computer subsystem which consists of a control processor (a host CPU with peripherals), and a high-speed arithmetic processor (an array processor or a bank/grid of processing elements)

2. a data storage subsystem

3. an image/data display subsystem

4. a microwave link subsystem to couple SARCOM to Gilmore Creek

The LO-RES SAR data can be handled by the SARCOM control processor and high-speed bus at real-time rates without exceeding the 1.33 Mbps maximum satcom link rate and without being compressed by the arithmetic processor. The HI-RES SAR data will be handled by SARCOM at real-time rates by utilizing not only the control processor and high-speed bus but also the high-speed arithmetic processor to compress the data by compression ratios as high as 30-to-1 or 45-to-1 at real-time processing rates to fit within the 1.33 Mbps maximum.

A data storage subsystem consists of a high-speed, high-capacity disk drive and its interface to the real-time input/output bus. The data storage subsystem can provide for interim storage of a single image frame or as a stacker of image frames prior to being forwarded to the satcom link at Gilmore Creek.

The seven functional components of the SARCOM end-to-end system in Alaska and in Maryland are indicated in Figure 5.

Figure 6 shows a more detailed system interconnections diagram between the SARCOM and the main ASF systems. It includes four of the seven SARCOM functional components contained at the UAF.

There will be basically only two interconnections between the two systems. The critical LO-RES and HI-RES Quick Turnaround Data interface between the ASF Digital Cassette Recorder System (DCRS) and the SARCOM high speed Input/Output bus is depicted by the wide path running parallel to the bottom of Figure 6. The only other system interconnect will be a low speed network connection, designated DECNET, which will couple the SARCOM CPU and two of the ASF CPUs.

A preliminary design of the hardware subsystems which will achieve the SARCOM objectives is indicated in Figure 7 which includes possible manufacturers and models. The rationale for the system design was to incorporate off-the-shelf hardware subsystems with significant existing subsystem software (drivers, high speed mathematics library and diagnostics) and already demonstrated successful interfacing between hardware subsystems wherever possible.

Several candidates for the hardware subsystem designated as the array processor have been identified. This subsystem has as its principal function the responsibility to perform the data compression of the HI-RES data in real-time. However, a cost-effectiveness trade-off study of these processor candidates has not been finalized yet. Apparently, the significantly high, sustainable arithmetic load of 160-200 MFLOPS required for the most demanding modes of SARCOM hovers around the present technological limit of moderately priced ($0.25M-$0.50M) array processors.

SARCOM DATA COMPRESSION TECHNIQUES

A consideration of the various modes of operation for the three spaceborne SAR systems for which the SARCOM was designed establishes a range of design compression ratios which set the framework for the selection of a data compression algorithm(s). The real-time SARCOM design compression ratios for the extreme swathwidths and quantization levels (bits/sample) selectable for the three systems are given below. The values derived are based on a maximum link bandwidth of 1.33 Mbps and a pixel spacing of 12.5 meters.

SWATHWIDTH (KM)	BITS/SAMPLE	REAL-TIME IMAGE RATE (Mbps)	DESIGN COMPRESSION RATIO
150	8	61.4	46.2
100	8	41.0	30.8
50	4	10.4	7.7

Therefore, the range of SARCOM design compression ratios is from approximately 8-to-1 to approximately 30-to-1 for the E-ERS-1 and J-ERS-1 systems while the upper end of the range is extended to approximately 45-to-1 when the RADARSAT system is included.

Within this design compression ratio range, several data compression techniques were evaluated at the Digital Image Processing Laboratory (DIPL) using SEASAT SAR and Shuttle Imaging Radar (SIR-B) imagery of ice, land and sea[5]. Some of the techniques evaluated in both non-adaptive and adaptive forms included both, a) spatial techniques such as the linear, bi-linear and quadratic interpolative techniques, a linked polynomial technique and block truncation coding (BTC), and b) transform techniques such as the discrete cosine transform (DCT) and the Hadamard Transform (HT) techniques. The evaluation criteria used were, a) the fidelity of the reconstructed image determined by the polling of subjective viewing and the quantitative measure of normalized-mean-square-error (NMSE), and b) the arithmetic burden imposed on the real-time SARCOM system.

The results of the study indicated that for the SARCOM data compression range of 8-to-1 through 45-to-1 the two transform techniques, the DCT followed by the HT, yielded the best results. Even though the DCT produced significantly better fidelity, the HT method showed potential due to the its arithmetic simplicity of being reducible to just a number of additions and no multiplications. This simplicity of the HT method may be implemented very effectively in certain system architectures.

Figure 8 presents a block diagram of the data compression schematic for the Discrete Cosine Transform (DCT) technique at both the transmit and receive ends. The schematic is a generic one which is actually applicable to any transform technique. The indicated steps in this data compression process for the non-adaptive DCT technique (or any non-adaptive transform technique) are the well-known steps for any non-adaptive transform technique[6].

In a real-time data compression process the arithmetic load of any technique can be just as important as the effectiveness of that technique in producing the highest-fidelity reconstruction of the original image after data compression.

The arithmetic load of a data compression process on the transmit end can be characterized by the figure of merit given by the number of arithmetic operations (real adds and real multiplies) required to implement the data compression algorithm. In the DIPL study, the DCT technique was selected as the best performance non-adaptive technique based on spaceborne SAR ice imagery from SEASAT when both the arithmetic load and the effectiveness of the reconstructed image were considered.

Figure 9 presents plots of the number of additions and the number of multiplications per input pixel of the scene as a function of the compression factor are plotted for the non-adaptive DCT data compression technique. The curve parameter, N, in the figure is the block size of the square used as the subscene size for the application of the technique. For example, using a subscene size of 128-by-128 pixels as a block size, the non-adaptive DCT technique requires approximately 11 additions per input pixel and approximately 4 multiplications per input pixel or a total of 15 operations per input pixel for the most demanding SARCOM compression factors of 30-to-1 and 45-to-1. It is interesting to note that the number of operations per input pixel is essentially independent of the compression factor above a compression factor of approximately 10-to-1 for a fixed block size. Furthermore, for a constant compression factor the number of additions and multiplications per input pixel for the whole scene and thus the total arithmetic load increases as a slow function of N, the block size of the subscene used.

The actual selection of a subscene block size is usually a trade-off between choosing a low value of N to reduce the total arithmetic load and choosing a higher value of N to reduce the subscene block edge effects when the subscenes are mosaicked together to form the full scene.

The use of an adaptive technique was found to significantly increase the fidelity of the reproduced image after data compression for SAR ice imagery. For the SARCOM range of compression factors the arithmetic load imposed by the adaptive DCT technique could be bounded by 20-25 operations per input pixel for compromise subscene block sizes of N=128. This was the arithmetic load which was used in the design criteria for the SARCOM real-time operations.

The non-adaptive DCT algorithm selected is a combination of, a) a variation of the fast DCT algorithm of B.G. Lee[4] to achieve the forward (or inverse) DCT, and b) decision criteria developed at the DIPL based on the scene statistics to produce the actual data compression.

The adaptive DCT algorithm selected is a combination of, a) the same variation of Lee's fast DCT method to perform the forward and inverse transforms, and b) the very effective technique of Chen and Smith[2,3] implemented to accomplish the actual data compression.

An example of the very effective adaptive discrete cosine transform (ADCT) technique is displayed in Figure 10. It is an example of the reconstructed image (compression factor = 32) side-by-side with the original at a map scale of 1:500,000. The scene is a SEASAT SAR ice image of a portion of the Beafort Sea in the Arctic basin with Banks Island occupying the left one quarter of the image from top to bottom and the remainder of the image being pack ice near the island and much more mobile ice in the right half of the image. Figure 11 emphasizes the fidelity of the reconstructed image by presenting a 64-to-1 zoom of the previous figure with a more demanding map scale of 1:64,000. It is a blow-up of just a small portion of the ice floe region just to the right of image center in the complete scene.

The author gratefully acknowledges the innovative and efficient computer programming support provided by K.W. Hoppel in the data compression algorithm development and evaluation.

REFERENCES

1) Carsey, F., "Alaska SAR Facility (ASF) Science Requirements", Jet Propulsion Laboratory, JPL D-3668, January 1988.

2) Chen, W., C.H. Smith and S.C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Transactions on Communications, Vol. COM-25, pp. 1004-1009, September 1977.

3) Chen, W. and C.H. Smith, "Adaptive Coding of Monochrome and Color Images", IEEE Transactions on Communications, Vol. COM-25, pp. 1285-1292, November 1977.

4) Lee, B. G., "A New Algorithm to Compute the Discrete Cosine Transform", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp. 1243-1245, December 1984.

5) Mango, S.A., K.W. Hoppel, P.B. Bey and M.R. Grunes, "Case Study : Data Compression Techniques for Synthetic Aperture Radar (SAR)", Scientific Data Compression Workshop, Snowbird, Utah, May 1988.

6) Pratt, W. K., "Digital Image Processing", Wiley (Interscience), New York, 1978.

TABLE 1
SAR SENSOR MISSION DESCRIPTIONS
ALASKA SAR FACILITY (ASF)

		E-ERS-1	J-ERS-1	RADARSAT
SAR	Frequency	C-band	L-band	C-band
	Polarization	VV	HH	HH
	Swath	80km	75km	500 to 50 km
	Resolution/looks	30m/4	30m/4	100m/8 to 10m/1
	Incidence angle	23 degrees	35 degrees	20-50+ degrees
	Orientation	Right	Right	Right
	On-Board Storage	none	about 10^{11} bits	yes
Orbit	Inclination	97.5 Degrees	97.7 degrees	98.5 degrees
	Altitude	785 km	568 km	790 km
	Repeat	3 days initially, then TBD	41days	16 days (with 3-day subcycle)
	Node	sun-synchronous	sun-synchronous	sun-synchronous
Mission	Launch	4/1990	2/1992	1994
	Lifetime	2-3 years	2 years	5 years
	Status	Approved	Approved	Approved
Other Instruments		Radar Altimeter, Wind/Wave Scatterometer, Along-Track Scanning Radiometer	Optical Sensor	TBD

[FROM CARSEY, 1988]

ORIGINAL PAGE IS
OF POOR QUALITY

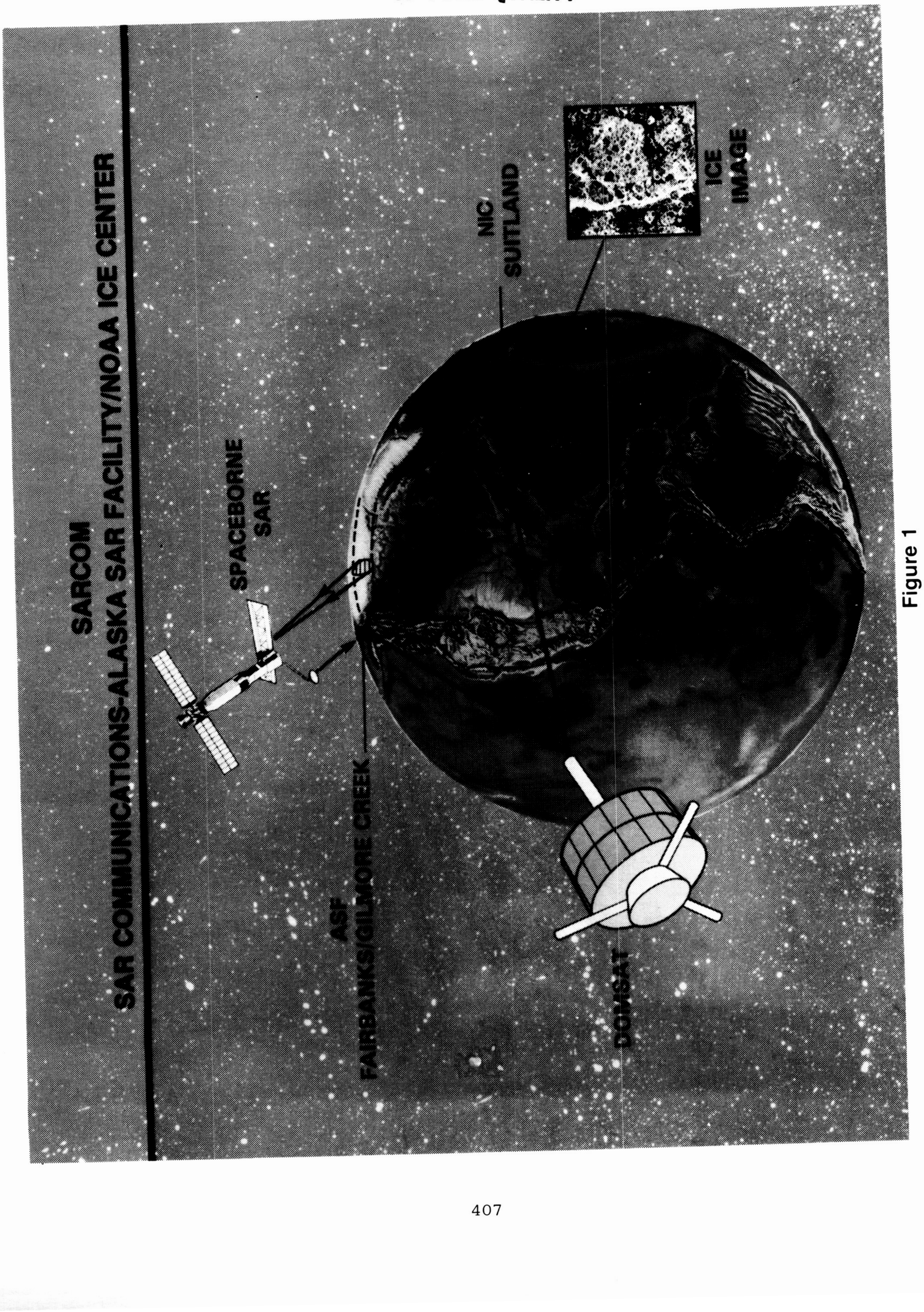
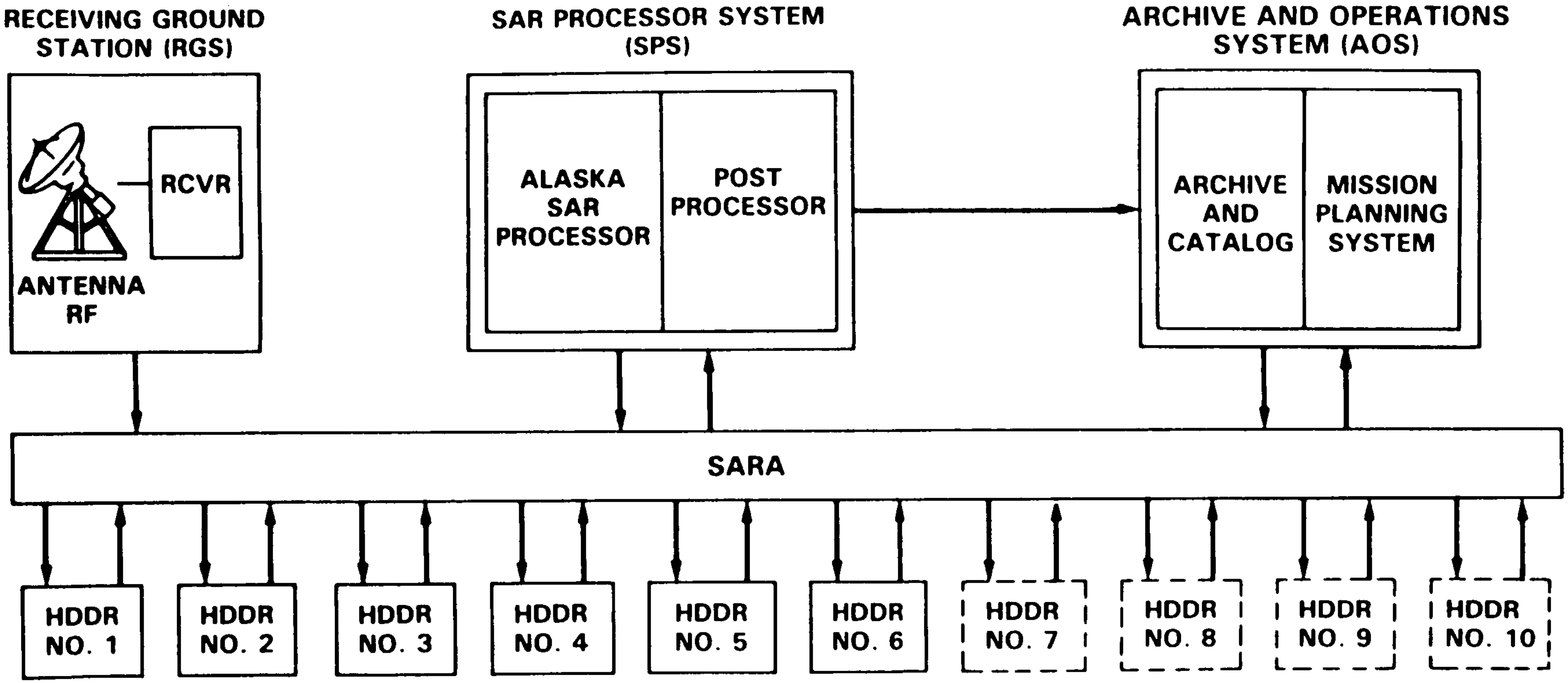


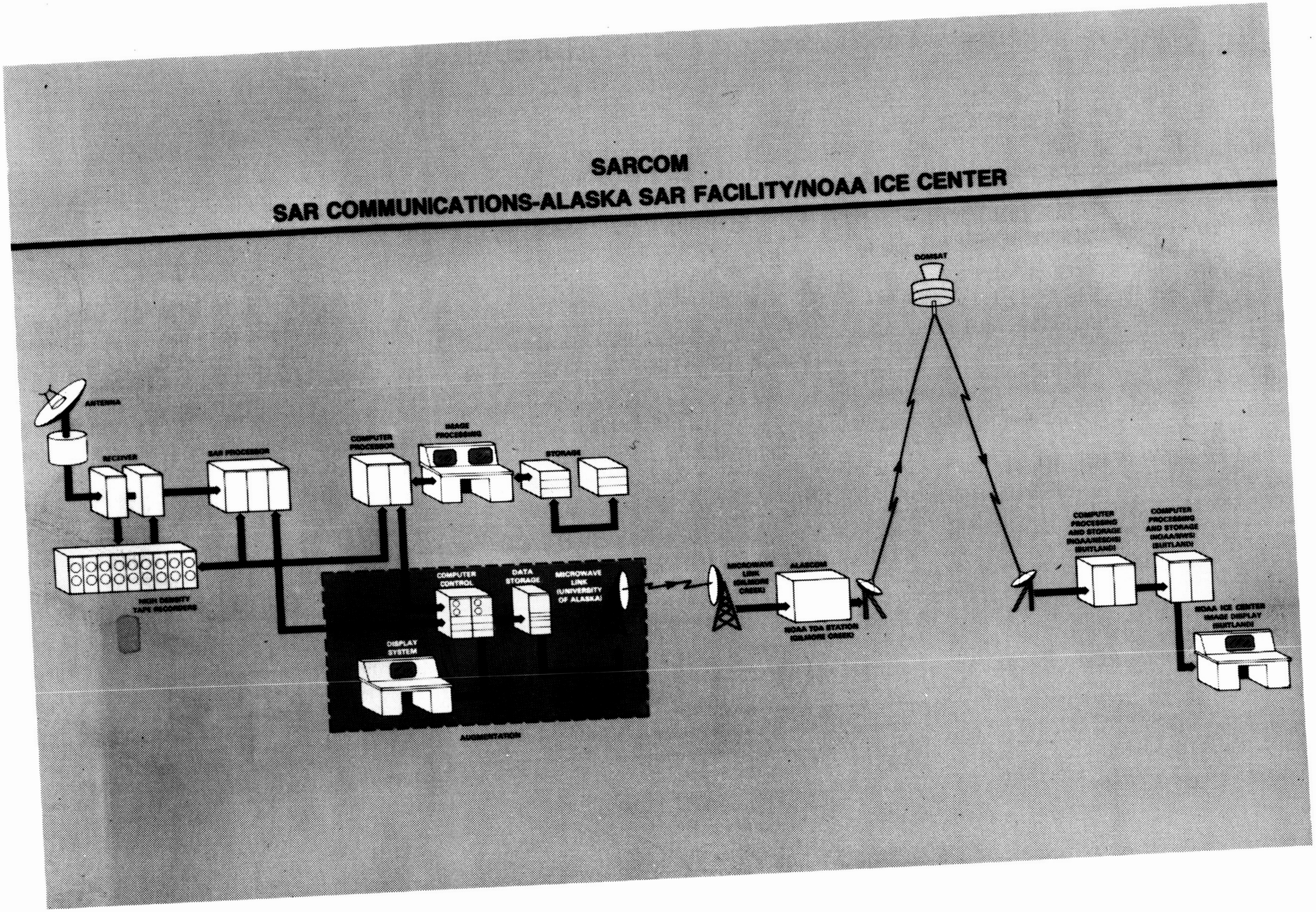
Figure 1

ALASKA SAR FACILITY (ASF) FUNCTIONAL BLOCK DIAGRAM



[FROM JPL ASF DESIGN REVIEW, JUNE 1987]

FIGURE 2



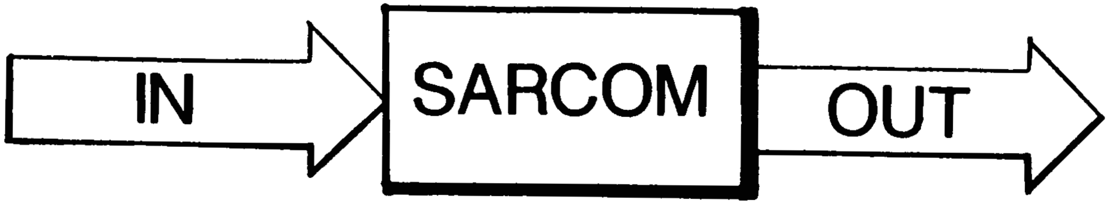
SARCOM

SAR COMMUNICATIONS-ALASKA SAR FACILITY/NOAA ICE CENTER

ANTENNA
RECEIVER
SAR PROCESSOR
COMPUTER PROCESSOR
IMAGE PROCESSING
STORAGE
HIGH DENSITY TAPE RECORDERS
COMPUTER CONTROL
DATA STORAGE
MICROWAVE LINK (UNIVERSITY OF ALASKA)
DISPLAY SYSTEM
AUGMENTATION
MICROWAVE LINK (GILMORE CREEK)
ALASCOM
NOAA TDA STATION (GILMORE CREEK)
DOMSAT
COMPUTER PROCESSING AND STORAGE (NOAA/NESDIS) (SUITLAND)
COMPUTER PROCESSING AND STORAGE (NOAA/NWS) (SUITLAND)
NOAA ICE CENTER IMAGE DISPLAY (SUITLAND)

Figure 3

SARCOM FUNCTION AT ASF



- LO-RES OR HI-RES
  QUICK-TURNAROUND DATA
  < 6 HOURS "OLD"
- MEDIUM-AMPEX DCRSi
  CASSETTES (AMPEX/JPL FORMAT)
- REAL-TIME RATE
  8 MBYTES/SEC MAX.
  1 BYTE/IMAGE PIXEL
  12,000 PIXELS/SWATHLINE MAX
  640 SWATHLINES/SEC MAX

- EFFECTIVE THRUPUT RATE
  8 MEGAPIXELS/SEC
- ARITHMETIC LOAD
  20-25 REAL OPERATIONS
  PER INPUT PIXEL
- EFFECTIVE ARITHMETIC RATE
  160-200 MFLOPS

- HI-RES (OR LO-RES)
  COMPRESSED REPRESENTATION OF IMAGE
  (OR UNCOMPRESSED IMAGE)
- DOMSAT/NOAA DATA INGEST
  FORMAT
- REAL-TIME RATE
  167 KBYTES/SEC MAX
  (1.33 Mbps)
  1 BYTE/COMPRESSED
  COEFFICIENT
  (OR UNCOMPRESSED
  IMAGE PIXEL)

FIGURE 4

ASF SARCOM FUNCTIONAL COMPONENTS

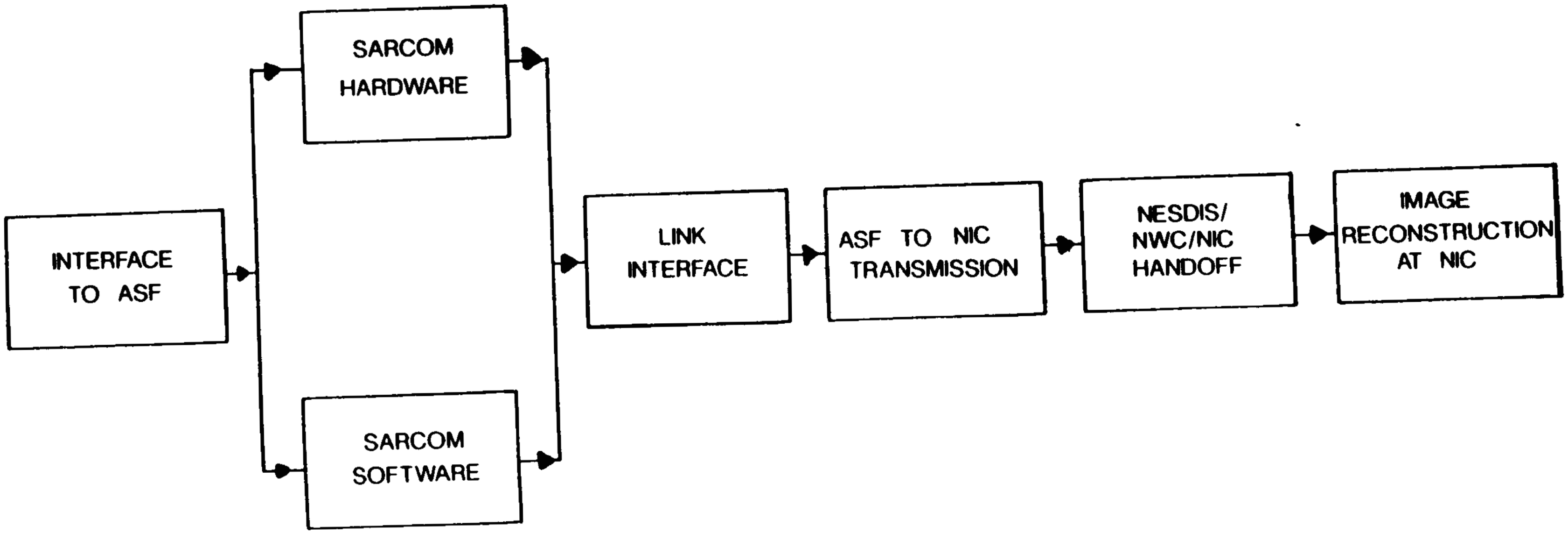


FIGURE 5

SYSTEM INTERCONNECTIONS

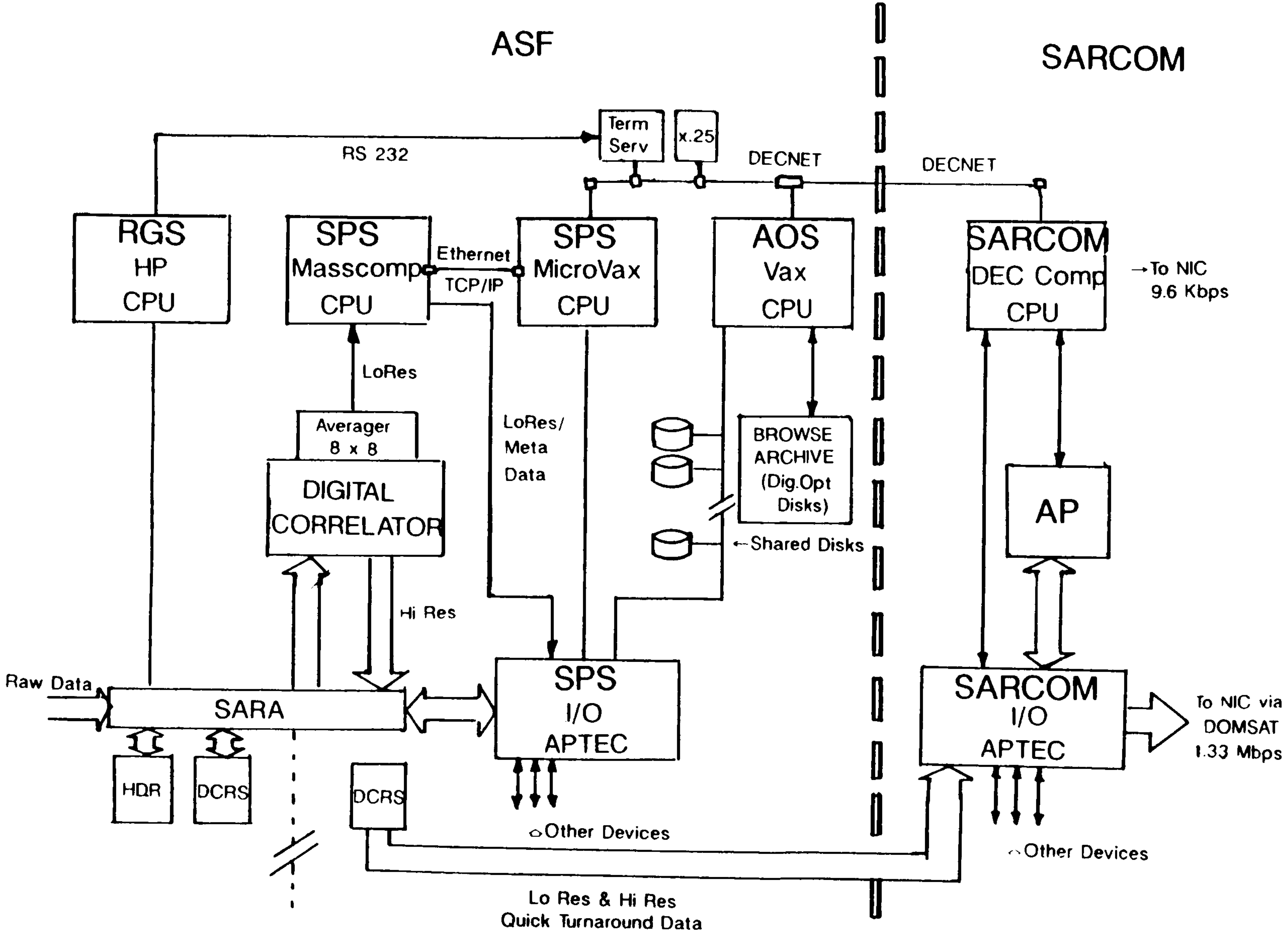


FIGURE 6

ASF SARCOM
HARDWARE SUBSYSTEMS

(BY MANUFACTURER/MODEL)

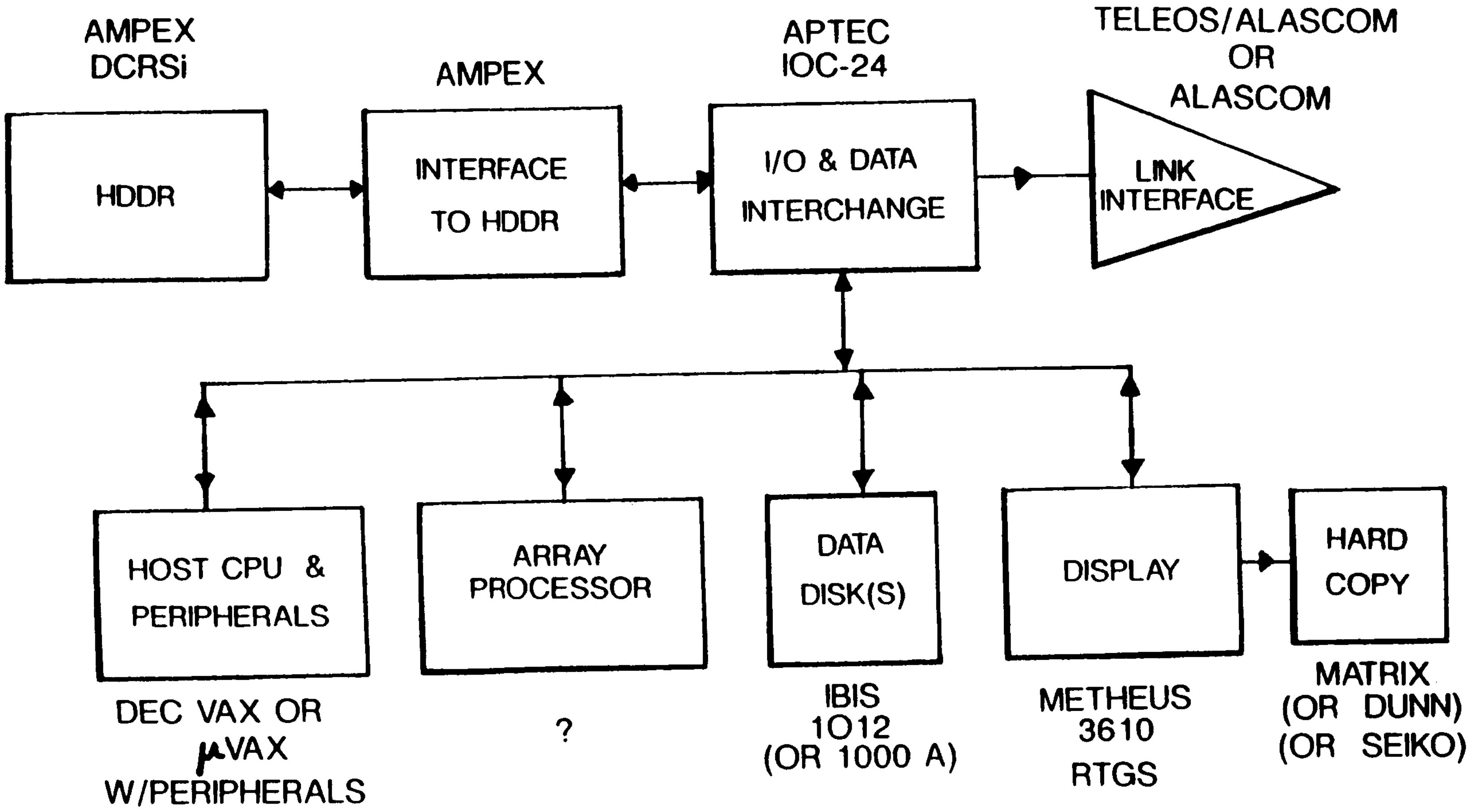
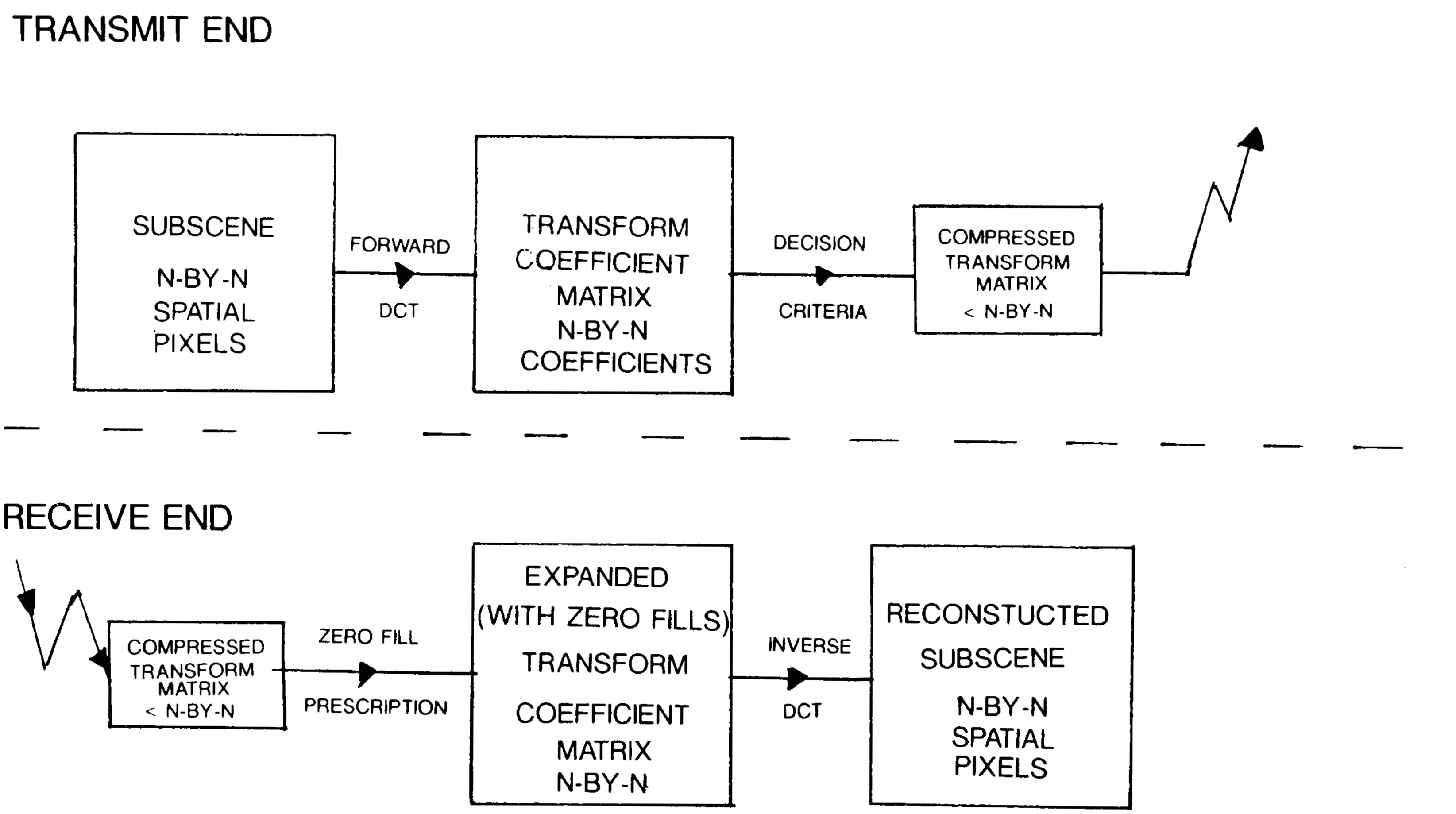


FIGURE 7

DATA COMPRESSION SCHEMATIC
FOR DISCRETE COSINE TRANSFORM (DCT) TECHNIQUE



TRANSMIT END

SUBSCENE
N-BY-N
SPATIAL
PIXELS

FORWARD
DCT

TRANSFORM
COEFFICIENT
MATRIX
N-BY-N
COEFFICIENTS

DECISION
CRITERIA

COMPRESSED
TRANSFORM
MATRIX
< N-BY-N

RECEIVE END

COMPRESSED
TRANSFORM
MATRIX
< N-BY-N

ZERO FILL
PRESCRIPTION

EXPANDED
(WITH ZERO FILLS)
TRANSFORM
COEFFICIENT
MATRIX
N-BY-N

INVERSE
DCT

RECONSTUCTED
SUBSCENE
N-BY-N
SPATIAL
PIXELS

FIGURE 8

SARCOM ARITHMETIC LOAD
NON-ADAPTIVE DISCRETE COSINE TRANSFORM (DCT)

N = BLOCKSIZE (N-BY-N PIXELS)

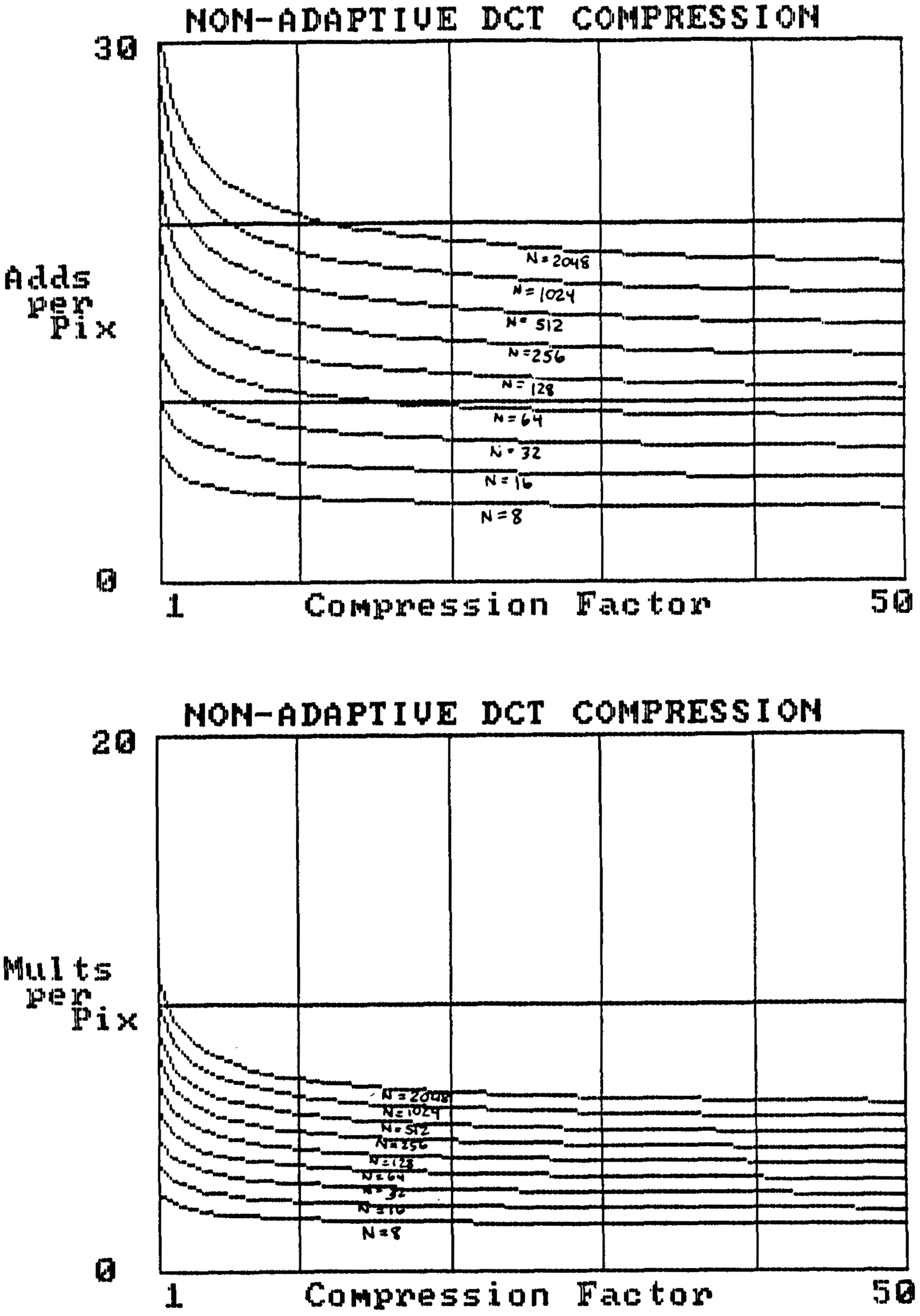
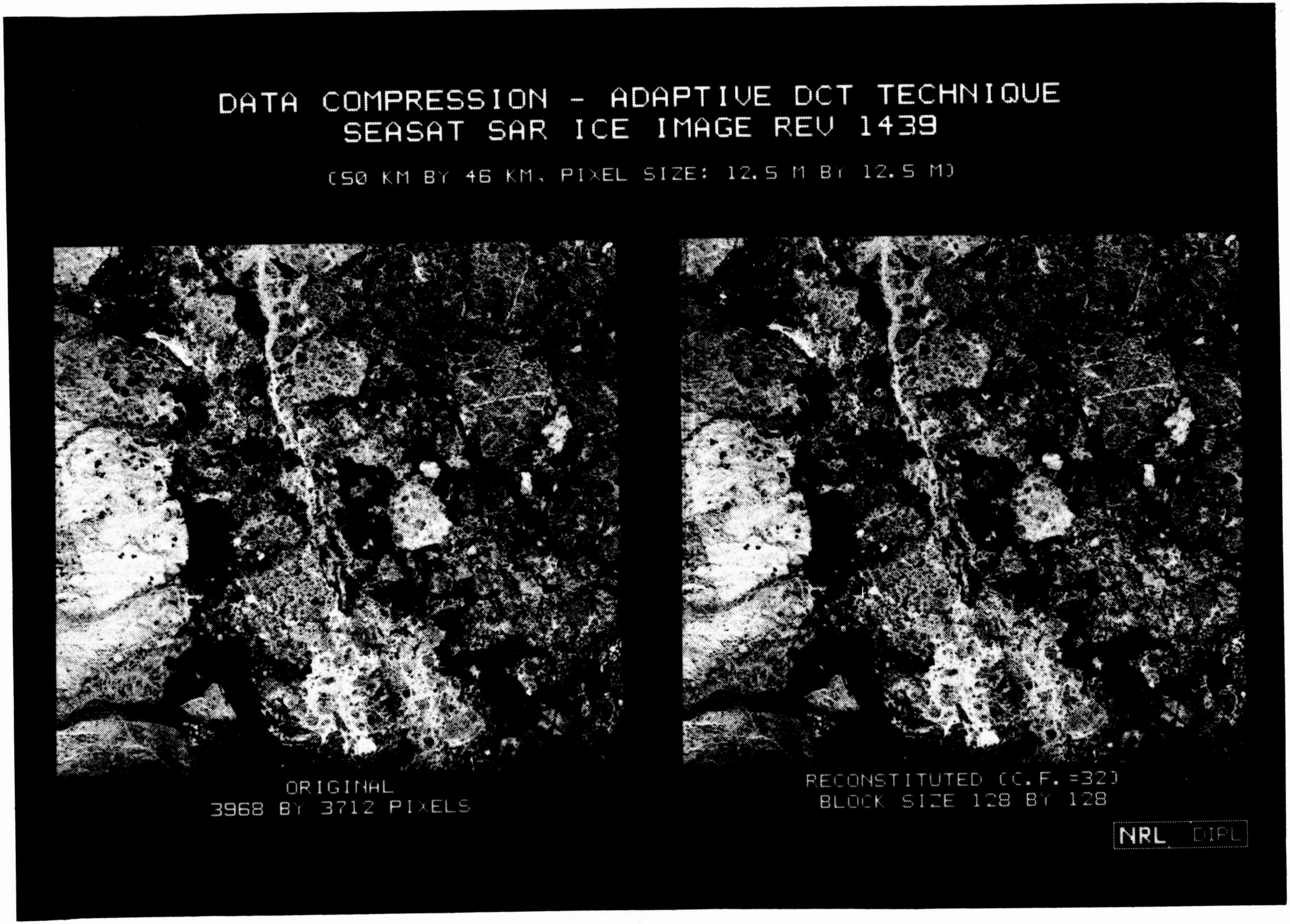


FIGURE 9



DATA COMPRESSION - ADAPTIVE DCT TECHNIQUE
SEASAT SAR ICE IMAGE REV 1439

(50 KM BY 46 KM, PIXEL SIZE: 12.5 M BY 12.5 M)

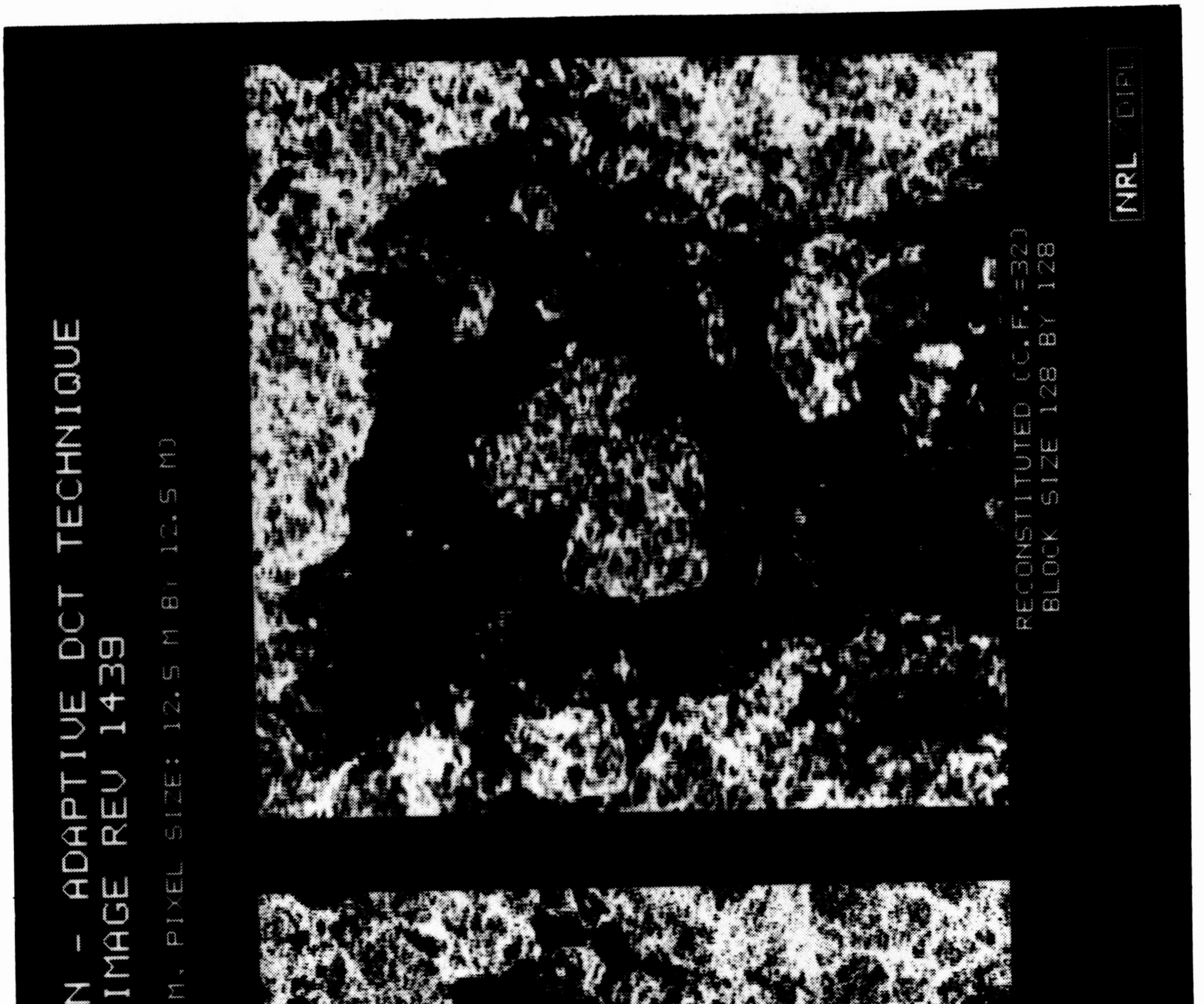
ORIGINAL
3968 BY 3712 PIXELS

RECONSTITUTED (C.F.=32)
BLOCK SIZE 128 BY 128

Figure 10

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY



DATA COMPRESSION - ADAPTIVE DCT TECHNIQUE
SEASAT SAR ICE IMAGE REV 1439

(6.4 KM BY 6.4 KM, PIXEL SIZE: 12.5 M BY 12.5 M)

Figure 11

A VLSI CHIP SET FOR REAL TIME VECTOR QUANTIZATION OF IMAGE SEQUENCES

Richard L. Baker
Integrated Circuits and Systems Laboratory
Department of Electrical Engineering
University of California

ABSTRACT

This paper describes the architecture and implementation of a VLSI chip set that vector quantizes (VQ) image sequences in real time. The chip set forms a programmable Single-Instruction, Multiple-Data (SIMD) machine which can implement various vector quantization encoding structures. Its VQ codebook may contain unlimited number of codevectors, N, having dimension up to K = 64.

Under a weighted least squared error criterion, the engine locates at video rates the best code vector in full-searched or large tree searched VQ codebooks. The ability to manipulate tree structured codebooks, coupled with parallelism and pipelining, permits searches in as short as O(log N) cycles. A full codebook search results in O(N) performance, compared to O(KN) for a Single-Instruction, Single-Data (SISD) machine. With this VLSI chip set, an entire video code can be built on a single board that permits realtime experimentation with very large codebooks.

OVERVIEW

- MULTISPECTRAL COMPRESSION PROBLEM
- PHILOSOPHY <---> A NEED
- VX IMPLEMENTATION CHALLENGES
- VQ CHIP SET

COMPRESSION RESEARCH AT UCLA

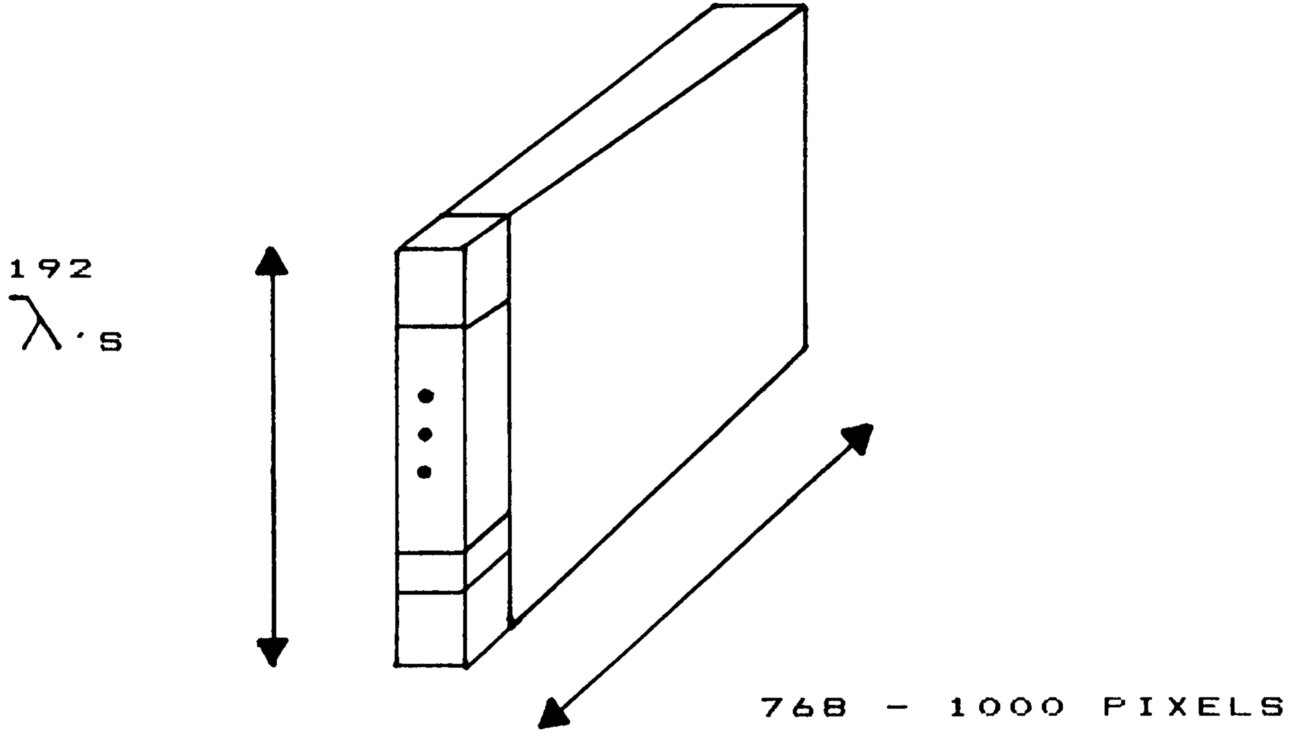
<u>ALGORITHMS</u>

- LOW RATE VIDEO
- SINGLE FRAME
- MULTISPECTRAL

<u>HARDWARE</u>

- APPLICATION SPECIFIC INTEGRATED CIRCUITS

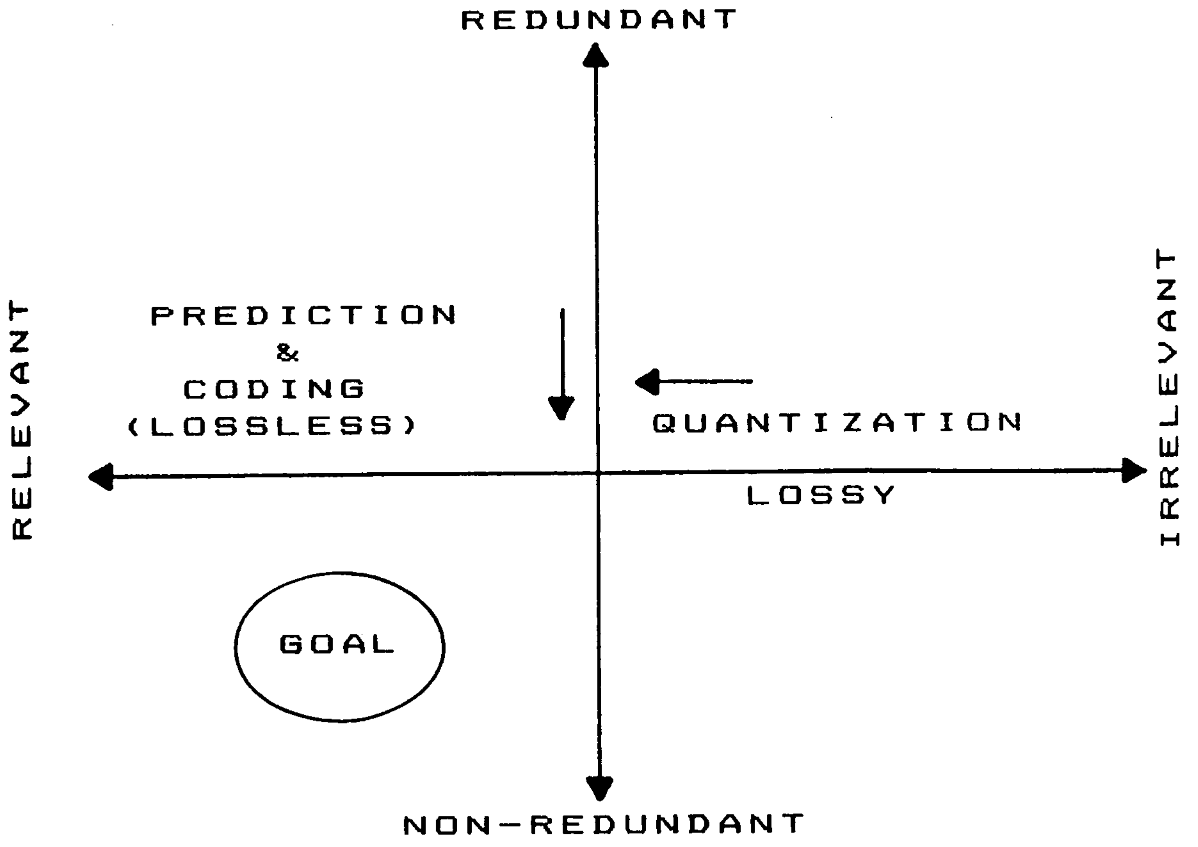
MULTISPECTRAL COMPRESSION PROJECT (JPL)



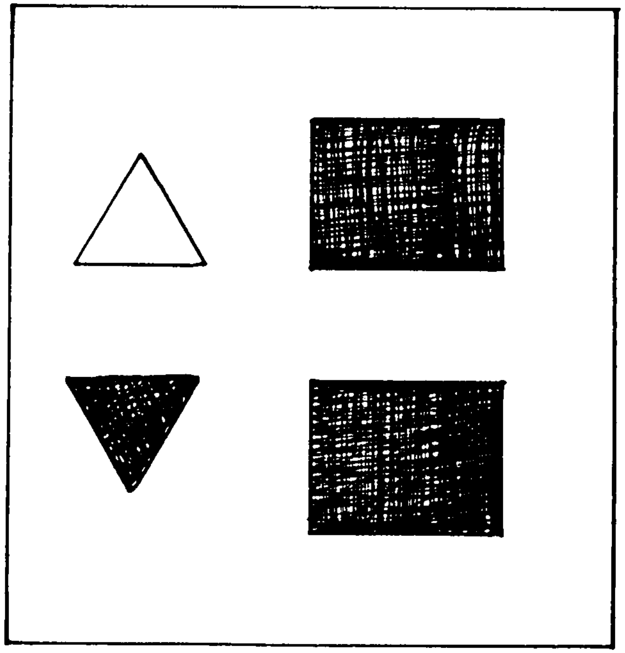
$$\frac{\text{50 MEGAPIXELS/SEC}}{\text{17 CHANNELS}} = \text{3 X } 10^{6} \text{ PIX/SEC/CH}$$

DESIRE OVER 50:1
--> UNDER 1/4 BITS/PIXEL

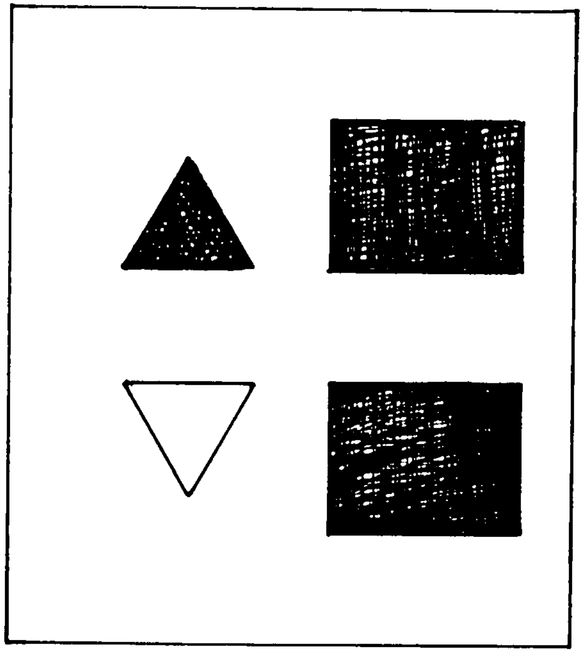
DATA COMPRESSION PROBLEM



DESIGNER'S PERCEPTIONS
VS.
USER'S NEEDS



CONFERENCE LEVEL



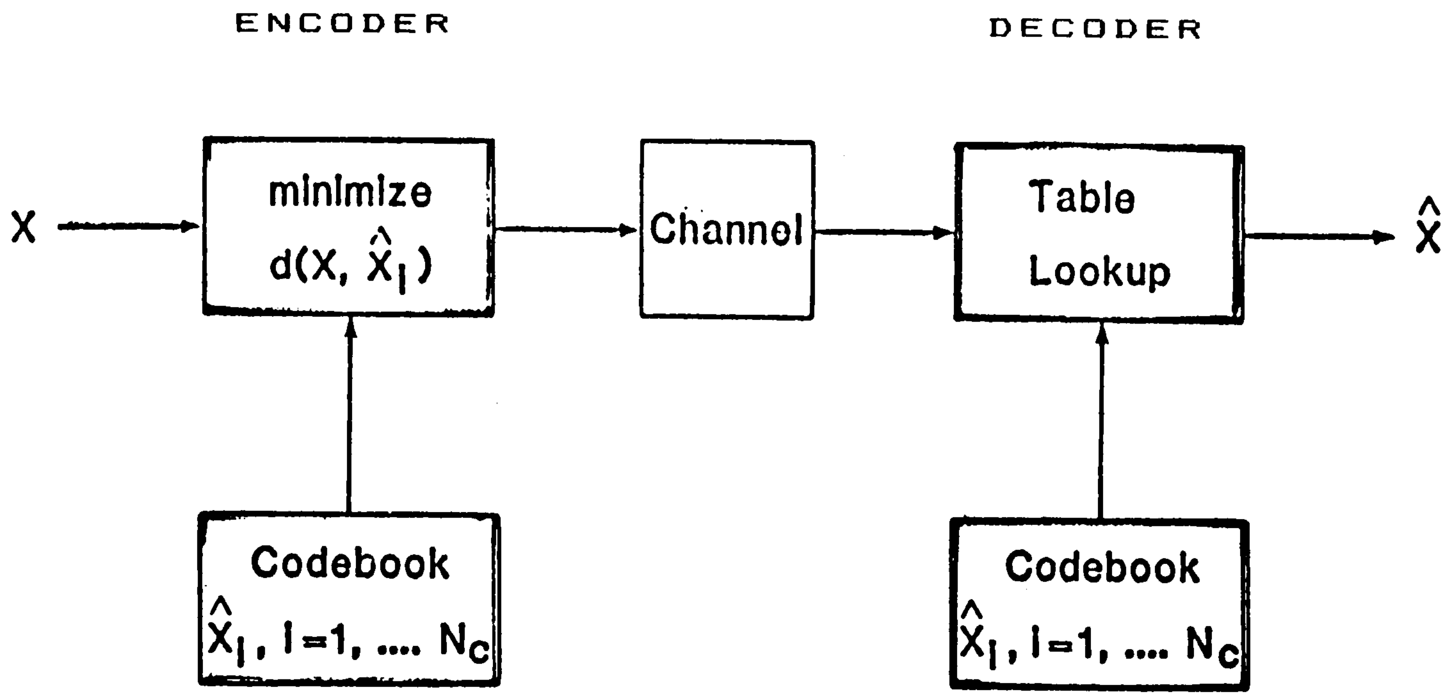
9TH FLOOR

WHAT IS RELEVANT?

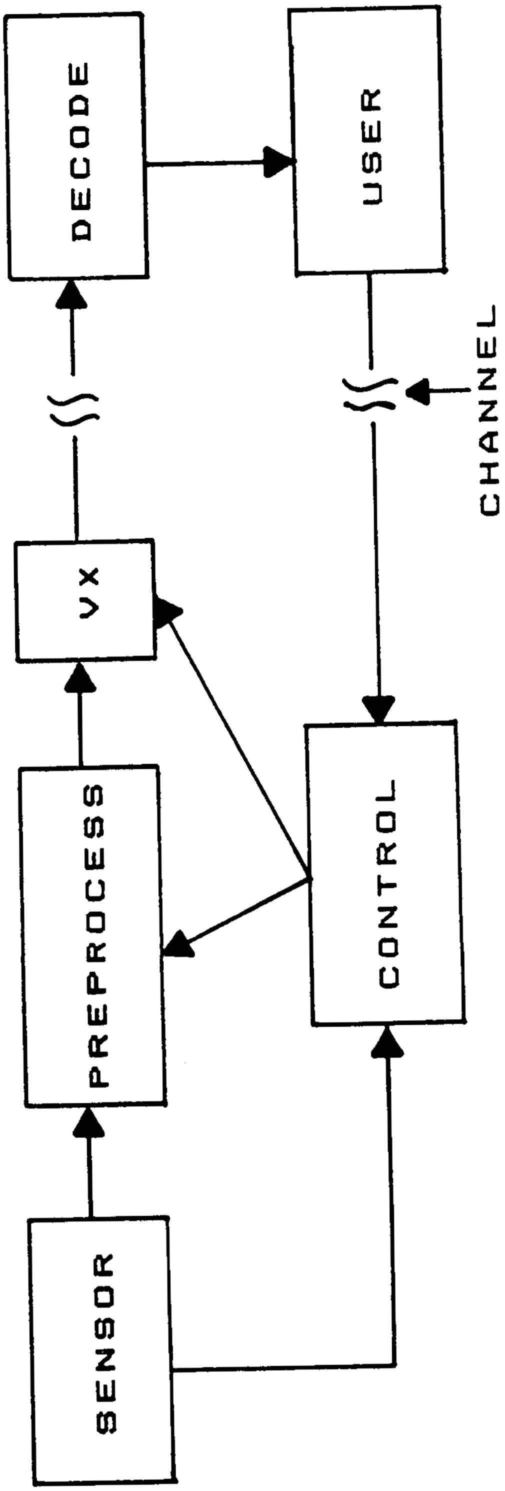
WHAT IS REAL?

DEPENDS ON USER

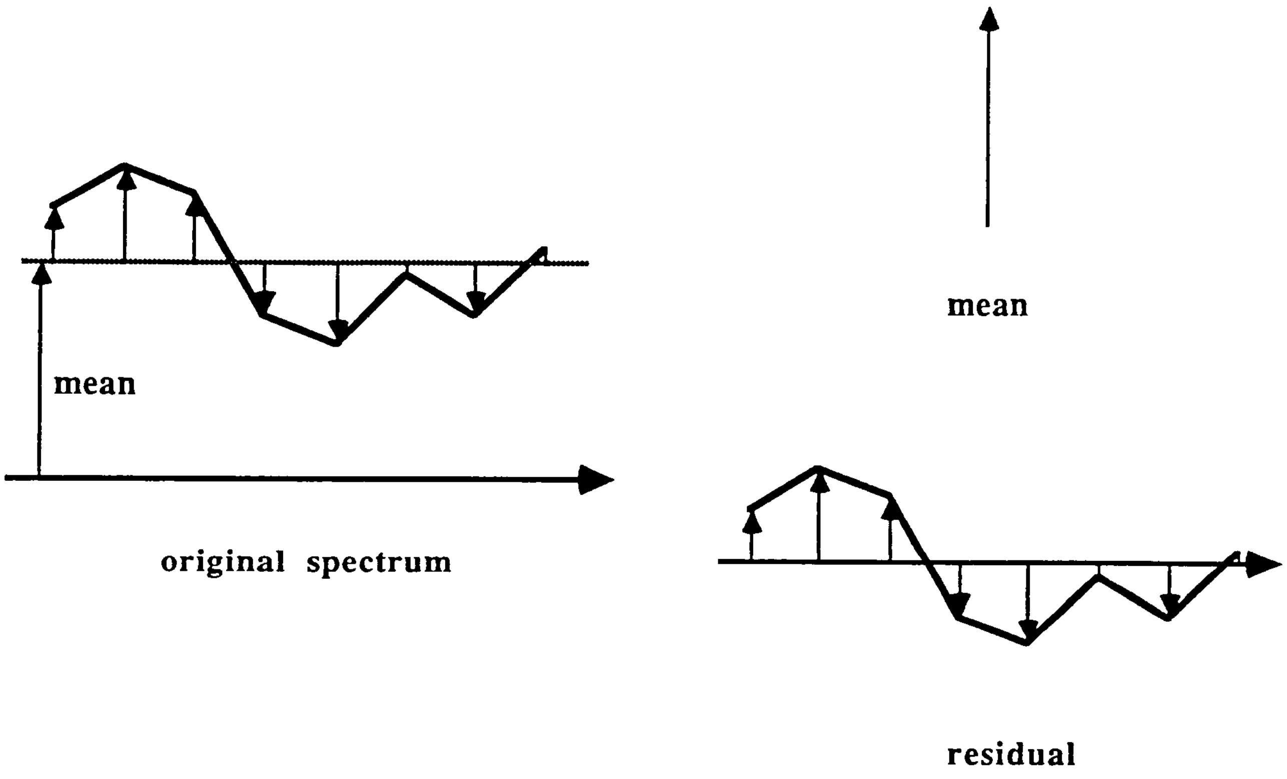
- MEAN SQUARE ERROR
- HAUSDORFF MEASURE
- HUMAN VISION SYSTEM MODELS
- MISSION SCIENTIST MODELS


ENCODER
DECODER
X
minimize
$d(X, \hat{X}_l)$
Channel
Table
Lookup
$\hat{X}$
Codebook
$\hat{X}_l, l=1, ... N_c$
Codebook
$\hat{X}_l, l=1, ... N_c$

Basic VQ


SENSOR
PREPROCESS
VX
DECODE
USER
CHANNEL
CONTROL

Mean-Residual VQ Encoder (MRVQ)

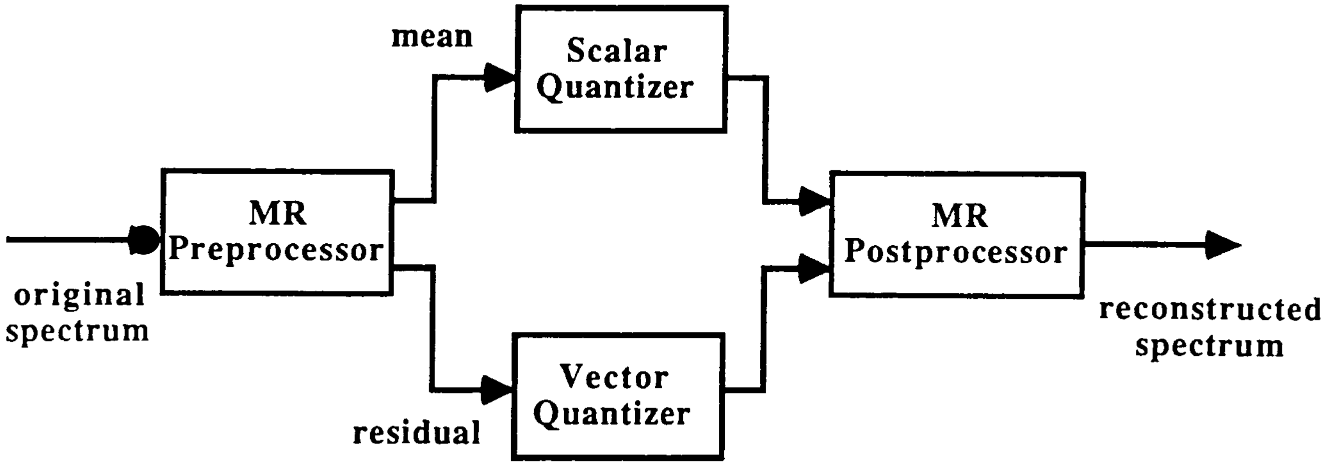


mean

original spectrum

mean

residual



mean

Scalar Quantizer

MR Preprocessor

original spectrum

MR Postprocessor

reconstructed spectrum

residual

Vector Quantizer

DISTORTION COMPUTATION

Minimize squared error:

$\mathbf{x}$ = Source vector, $\hat{\mathbf{x}}^i$ = ith Code vector,

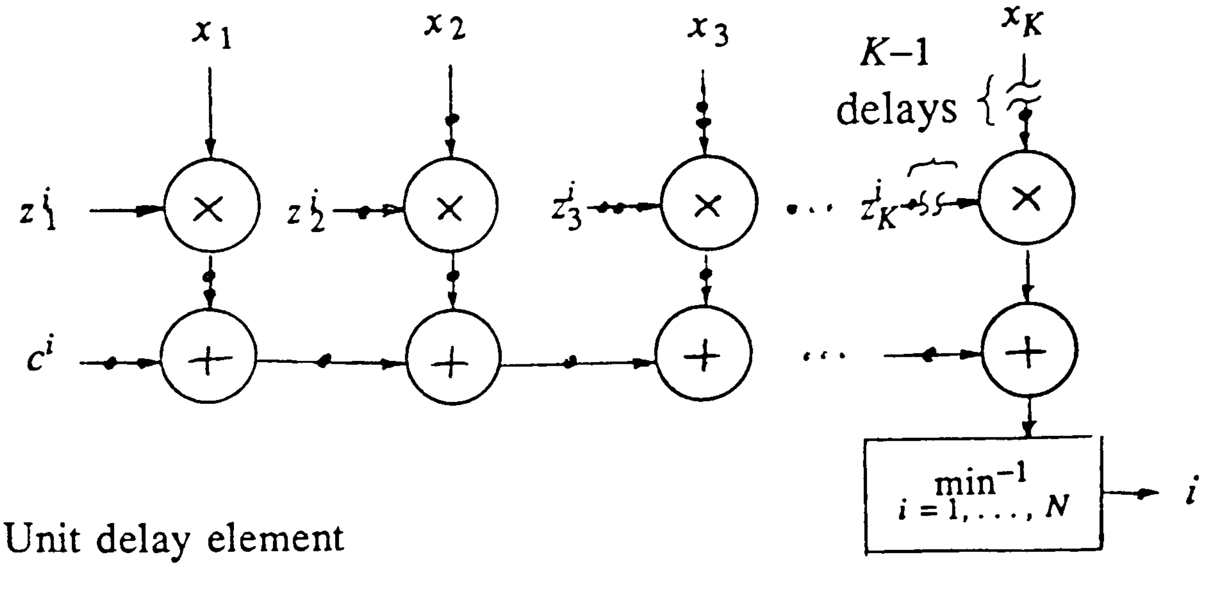
$$i = \min_{i=1,\ldots,N}{}^{-1} \left\{ \sum_{k=1}^{K} w_k \mid x_k - \hat{x}_k^i \mid^2 \right\},$$

$$= \min_{i=1,\ldots,N}{}^{-1} \left\{ \sum_{k=1}^{K} \frac{w_k (x_k)^2}{2} - \sum_{k=1}^{K} w_k \hat{x}_k^i x_k + \sum_{k=1}^{K} \frac{w_k (\hat{x}_k^i)^2}{2} \right\},$$

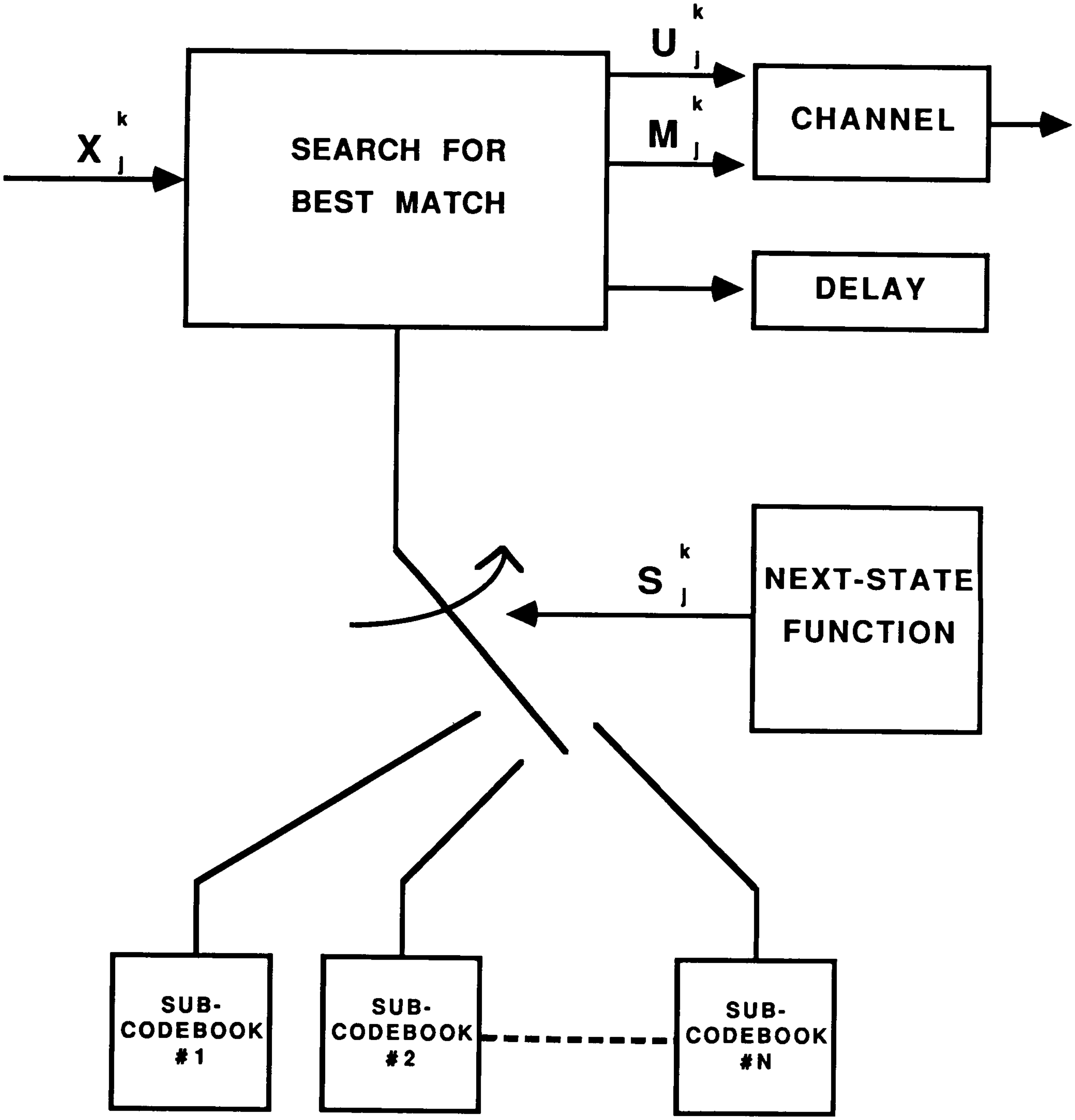
$$= \min_{i=1,\ldots,N}{}^{-1} \left\{ \sum_{k=1}^{K} z_k^i x_k + c^i \right\},$$

where

$$z_k^i \triangleq -w_k \hat{x}_k^i, \qquad c^i \triangleq \sum_{k=1}^{K} \frac{w_k (\hat{x}_k^i)^2}{2}.$$



● = Unit delay element



Basic Finite-State Vector Quantization Block Diagram.

PROBLEM: LIMITED SEARCH TIME

- Given:
  - 256x256 resolution image
  - 15 frames per second
  - 4x4 block size.

    → 983,040 pixels/sec

    → 61440 4x4 blocks/sec

  or 16.3 microseconds/block

- Assume:
  - Pipeline, 10 MHz clock, 1 distortion/clock

    → 163 distortion computations / block

    → 163 codevectors searched / block

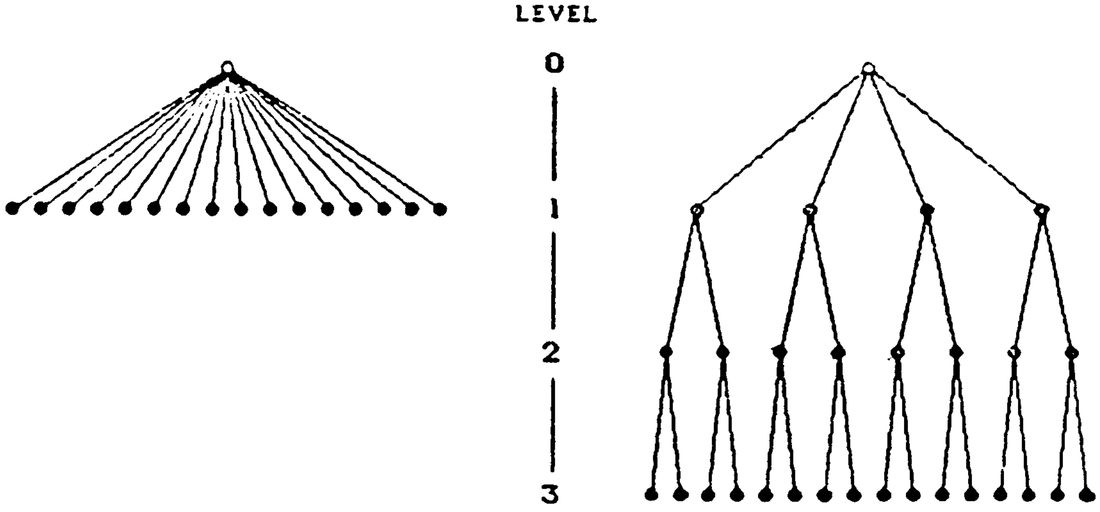
THESE #'S VARY AT RESOLUTION, BLOCKSIZE, RATE, ETC. - BUT:

- Problem:

  → Prefer 4000+ codevectors in codebook

  → Must limit search through codebook

ONE SOLUTION: TREES



N search
N memory

$O(\log N)$ search
$O(N)$ memory

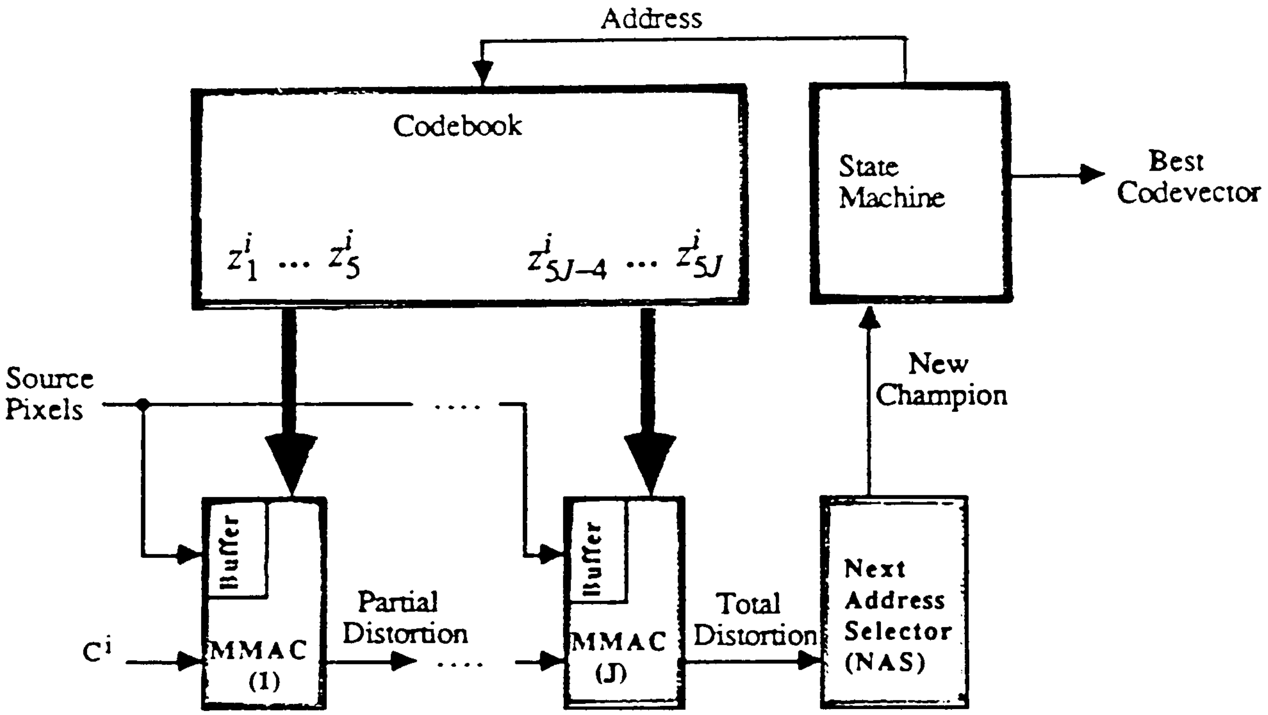
- Example

$$
\begin{aligned}
N &= 4096 = 2^{12} = 2^5 \times 2^7 \\
\text{Search} &= 2^5 + 2^7 = 160 \\
\text{Memory} &= 2^5 + 2^5 \times 2^7 = 32 + 4096 = 4128
\end{aligned}
$$

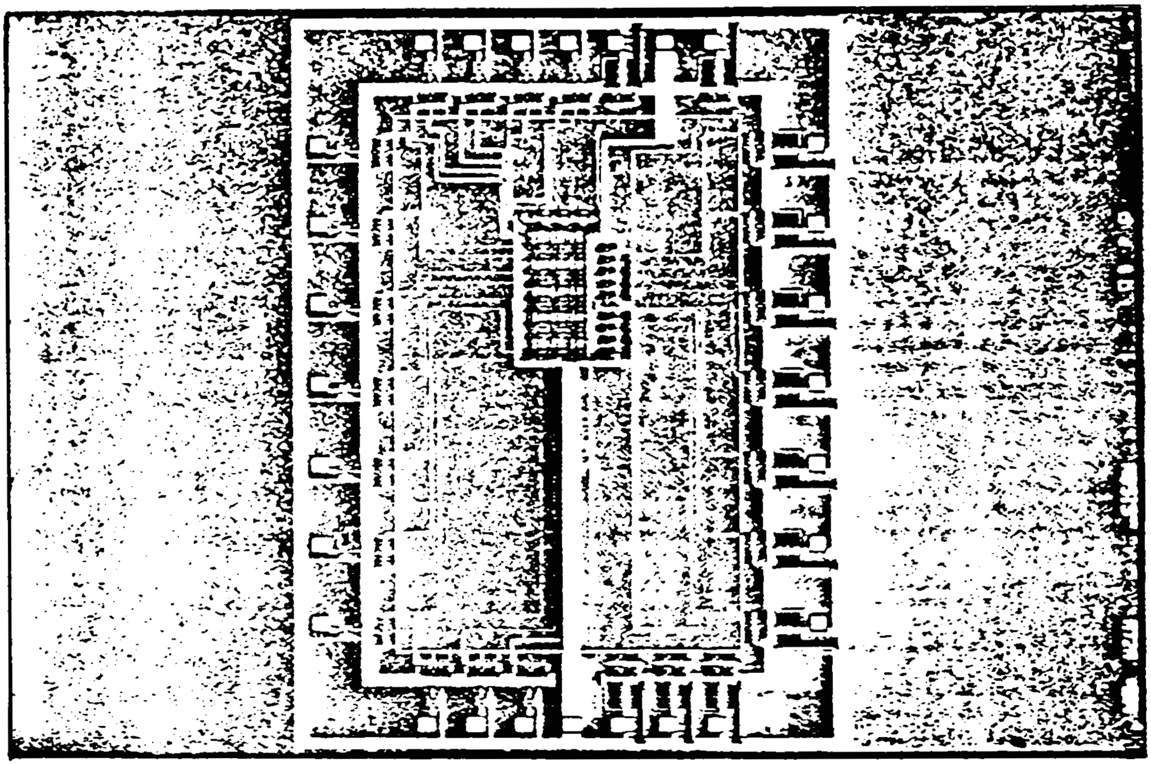
- Problem: data dependency

  - Minimize pipeline latency
  - Buffer to process several source vectors

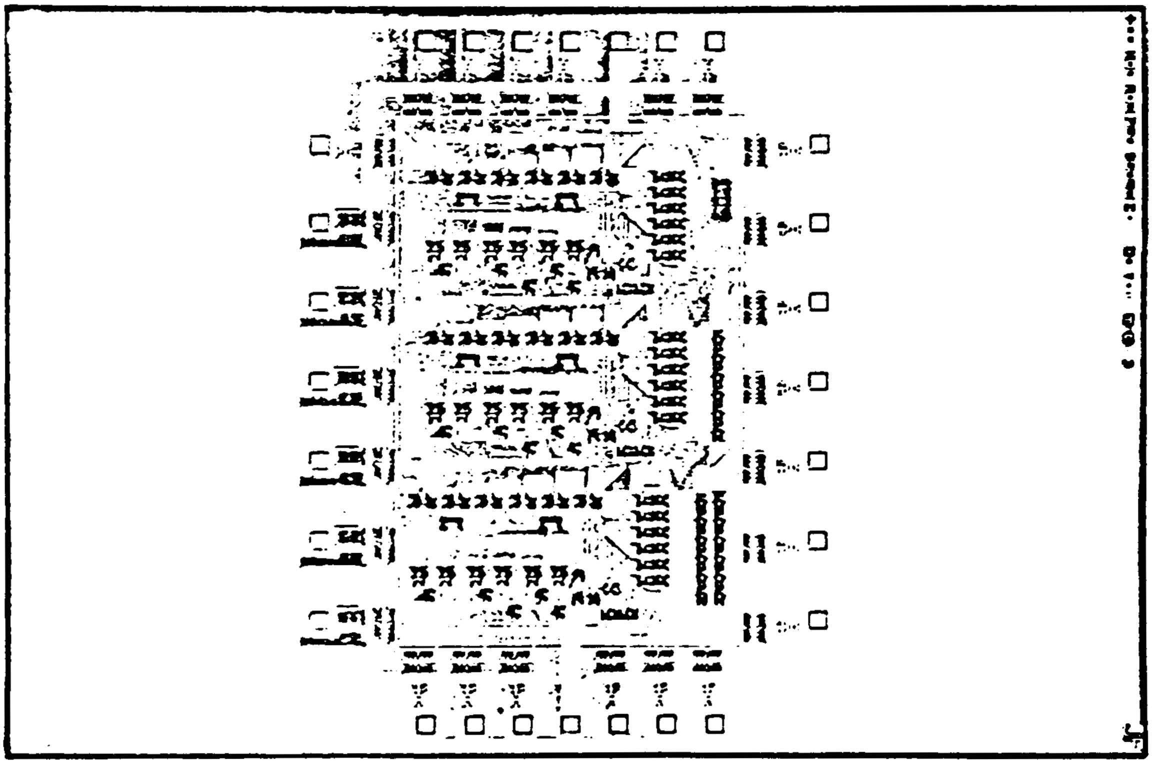
OVERALL SYSTEM



TALLY BLOCK (1/2)

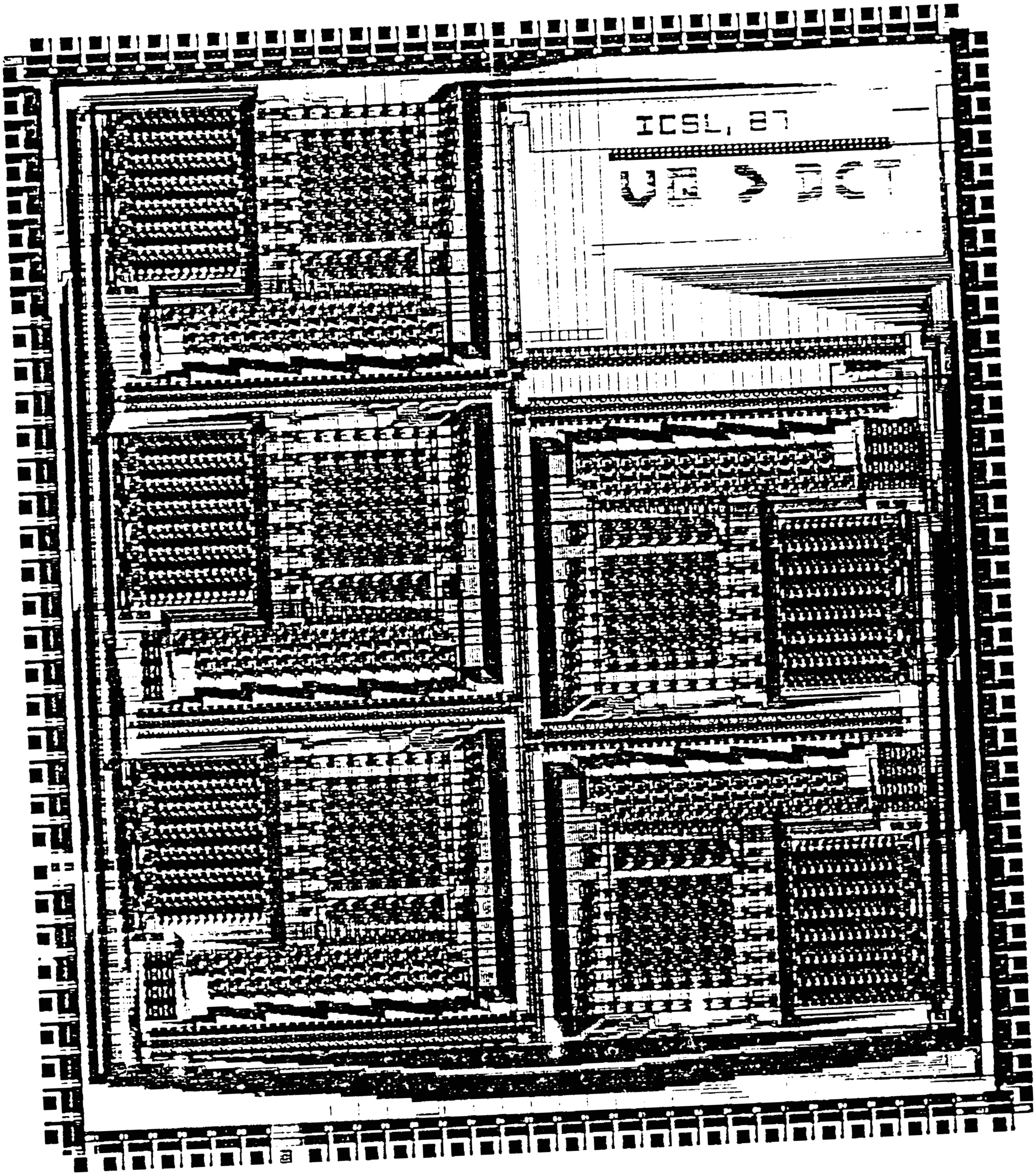


NEXT ADDRESS SELECTOR



ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY



FEATURES

- SEARCH TREE STRUCTURED CODEBOOKS
    - VECTOR DIMENSION UP TO 64 PIXELS
    - CODEBOOK SIZE LIMITED BY MEMORY
    - ONE DISTORTION COMPUTATION PER CLOCK
    - 6 BITS + SIGN
- ARCHITECTURE
    - SYSTOLIC ARRAY
    - ON CHIP BUFFERING
      --> FULL PROCESSOR UTILIZATION
    - CARRY SAVE ADDER AND DYNAMIC MANCHESTER CARRY CHAIN
    - PIPELINED COMPARATOR
- MMAC IMPLEMENTATION
    - 3 MICRON CMOS (MOSIS)
    - 7900 X 9200
    - ABOUT 30000 TRANSISTORS
    - 10 MHz PROJECTED => 10^7 VECTOR DISTORTIONS PER SECOND
    - 132 PINS
- NAS IMPLEMENTATION
    - 3400 X 4600
    - 1376 TRANSISTORS
    - 28 PIN
    - 12 MHz

SUMMARY

- MULTISPECTRAL COMPRESSION ALGORITHMS UNDER STUDY
- WHAT IS RELEVANT?
- HIGH SPEED VX CHIP SET
  - 10 MEGADISTORTIONS/SEC
  - TREE CODEBOOKS (LARGE)
  - INEXPENSIVE TECHNOLOGY

INTERNATIONAL STANDARDS ACTIVITIES IN IMAGE DATA COMPRESSION

Barry Haskell
AT&T Bell Labs

INTEGRATED SERVICES DIGITAL NETWORK (ISDN)

D channel ...... 16 kilobits/second (packetized)

B channel ...... 64 kilobits/second

H0 channel ...... 384 kilobits/second

H11 channel ...... 1.5 Megabits/second (T1)

H22 channel ...... 45 Megabits/second (T3)

H4 channel ...... 135 Megabits/second (packetized)

INTEGRATED SERVICES DIGITAL NETWORK (ISDN)

- Basic Access 2B + D
- Primary Access H11 = 23B + D
- Other Access (Evolving)

Coding for Color TV

Common Committee for International Radio (CCIR)

- Recommendation 601 (called CCIR 601)
  - Component coding - Y, Cr, Cb
  - Sampling: Y 13.5Mhz, Cr & Cb 6.75Mhz
  - Total bit rate 216 Megabits/second
  - Full frame: 720 x 480 NTSC, 720 x 576 PAL

Coding for Color TV

American National Standards Institute (ANSI)

- Committee T1Y1.1
  - So called "Network Quality"
  - NTSC Composite Signal Coding
  - Sampling: 14.32Mhz = 4 x Fsc
  - Bit rate: H11 T3 45Mbs
- Unofficial
  - So called "CATV" Quality
  - NTSC Composite Coding
  - Sampling: 10.7Mhz = 3 x Fsc
  - DPCM at 4 bits/pel
  - Bit rate: H22 ~ T3 ~ 45Mbs

  - Coding chips exist

Coding for Video Conferencing

Consultative Committee for Telephone and Telegraph (CCITT)

- Recommendations H.110 & H.120
  - Conditional Replenishment
  - Interframe DPCM
  - Bit rates: H11 ~ 1.5Mbs or H12 ~ 2.0Mbs

Coding for Video Conferencing

Consultative Committee for Telephone and Telegraph (CCITT)

- Recommendation H.12x
  - — Common Intermediate Format (CIF)
    - 360 pels, 288 lines, 30 frames/second, noninterlaced
  - — Conditional Replenishment
  - — Motion Compensation
  - — Discrete Cosine Transform (DCT)
    - DCT Chip available (8 x 8)
  - — Bit rates: N x 384 kbs (N x H0)
  - — Standard complete 1989

Coding for
Video Conferencing/Telephone

Consultative Committee for Telephone and Telegraph (CCITT)

- Recommendation H.????
  - Bit rates: M x 64 kbs
  - Conditional Replenishment
  - Motion Compensation
  - Remainder under study
  - Standard complete 1990???

Coding of Still Color Images

International Standards Organization (ISO)

Studied many algorithms

- Pel Domain
  - DPCM and Subsampling
  - Universal Coding
- Transform Domain
  - DCT fairly well understood
- Bit Plane Coding
  - Compatible with FAX

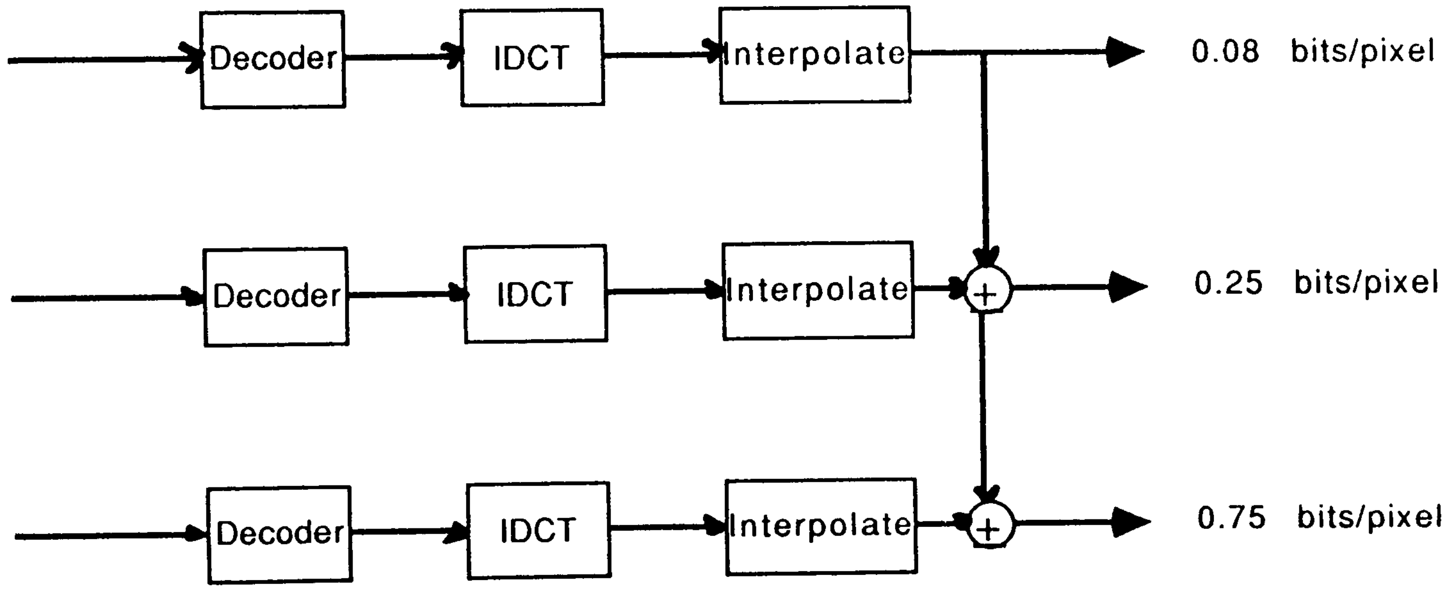
ISO Color Image Coding Standard

Pyramid DCT Gave Best Quality

- Progressive Coding and Transmission
  - .08, .25, .75 and 2.25 bits/pel
  - One final transmission for bit preservation

Proposal

ISO Still Picture Standard



Decoder → IDCT → Interpolate → 0.08 bits/pixel

Decoder → IDCT → Interpolate → + → 0.25 bits/pixel

Decoder → IDCT → Interpolate → + → 0.75 bits/pixel

Proposal

ISO Still Picture Standard

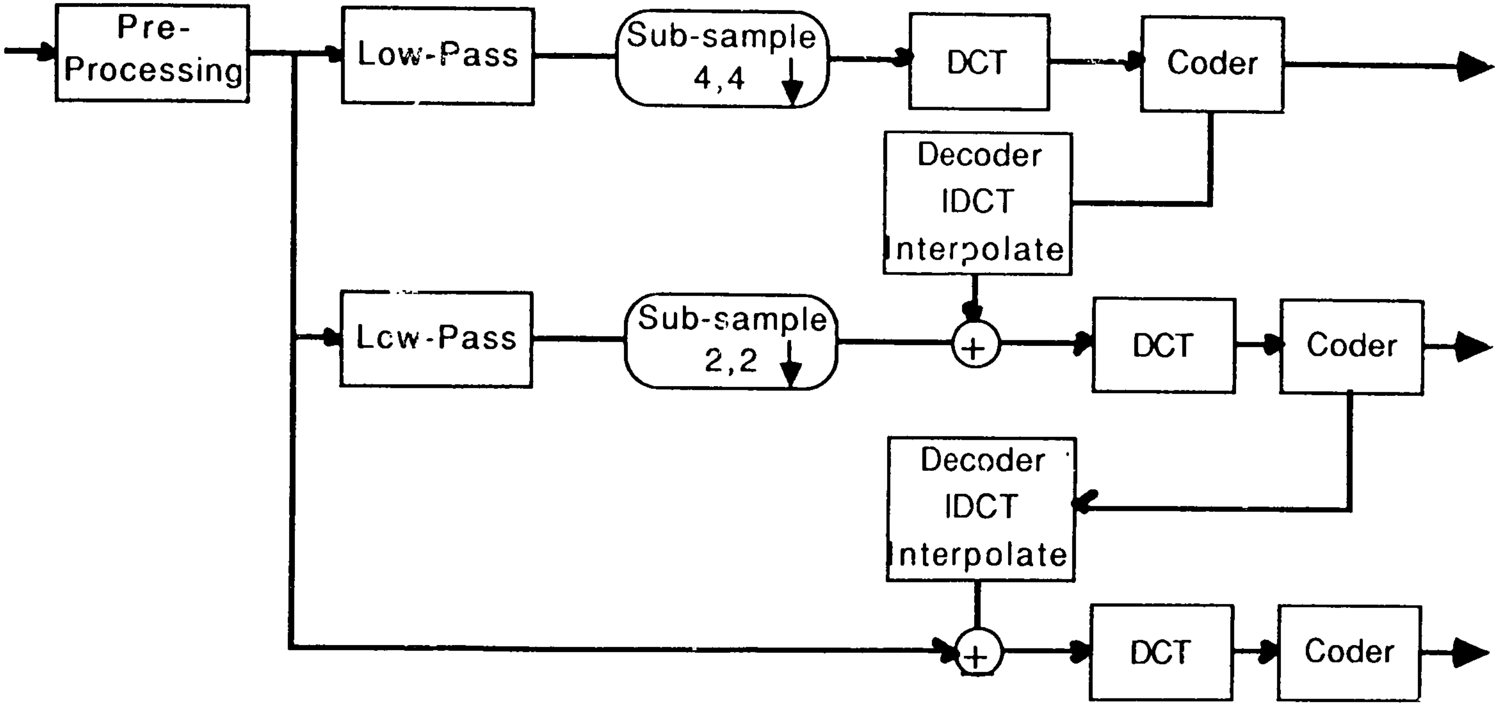


IMAGE PROCESSING USING GALLIUM ARSENIDE (GaAs) TECHNOLOGY

Warner H. Miller
NASA/Goddard Space Flight Center

ABSTRACT

The need to increase the information return from space-borne imaging systems has increased in the past decade. The use of multi-spectral data has resulted in the need for finer spatial resolution and greater spectral coverage. Finer spatial resolution and a greater number of spectral bands has increased data rated and system bandwidth requirements. Although the telecommunication capability planned through the 1900's is relatively large, feasibility studies on solid state imaging instruments in support of A1OS have shown increased rates that exceed the telecommunication channel capacity. Onboard signal processing will be necessary in order to utilize the available Tracking and Data Relay Satellite System (TDRSS) communication channel at high efficiency.

A generally recognized approach to the increased efficiency of channel usage is through data compression techniques. The method selected must function in real time satisfying the requirements of both the high speed data instrument source and limited bandwidth of the telecommunication channel. The compression technique implemented is a differential pulse code modulation (DPCM) scheme with a non-uniform quantizer.

NASA has recognized the need to advance the state-of-the-art of onboard processing and has chosen for this purpose to develop GaAs integrated circuit technology. NASA's GaAs research effort has developed an Adaptive Programmable Processor (APP) chip set which is based on an 8-bit slice general processor. This 8-bit slice and a control chip which stores the DPCM algorithm has been fabricated. This chip set will provide a compression ratio of 2 and operate in

real time to reduce a data rate from an imaging instrument to a data rate which is compatible with TDRSS.

The presentation will describe the reason for choosing the compression technique for the Multi-spectral Linear Array (MLA) instrument. Also. the presentation will give a description of the GaAs integrated circuit chip set which will demonstrate that data compression can be performed onboard in real time at data rate in the order of 500 Mb/s.

REAL TIME PROCESSING OF IMAGING ARRAYS
RELATIONSHIP BETWEEN SPATIAL RESOLUTION AND DATA RATE FOR ONE SPECTURAL BAND ASSUMING 185K FOV AND AN EARTH VIEWING ALTITUDE OF 700KM

	IFOV (M)	# DET. PER FOV	INTEG. TIME (DWELL) (MSEC.)	PIXEL RATE	PIXEL PERIOD (MICRO SEC.)	SERIAL DATA RATE PER SPECTRAL BAND (B/SEC.)
	120	1542	17.7	87 KPPS	11.5	695.7 K
MSS	80	2313	11.8	196K	5.1	1.57M
	60	3083	8.9	347.7K	2.9	2.78M
	40	4625	5.9	782.4K	1.3	6.26M
TM	30	6167	4.4	1.39Mpps	0.72	11.13M
	20	9250	2.9	3.13M	0.32	25.0 M
	15	12334	2.2	5.56M	0.179	44.5 M
ALOS/MLA	10	18500	1.47	12.5M	0.080	100.0 M

REQUIREMENTS

1. COMPRESSION RATIO $>$ 1.8
2. BE FABRICABLE FROM SPACE QUALIFIABLE PARTS
3. CONSUME LOW POWER, IMPLYING LOW ON-BOARD STORAGE
4. ALLOW FOR BIT TRANSMISSION ERRORS OF THE ORDER OF 1 IN 10^5
5. MINIMIZE SPATIAL AND RADIOMETRIC DISTORTION
6. MINIMAL PROCESSING POWER

ENTROPY MEASUREMENTS AND MAXIMUM
COMPRESSION RATIO WITH NO DISTORTION

	H(Y/X)	C_{MAX}
URBAN SCENE	3.283	2.44
NATURAL SCENE	3.051	2.62

THE INTERESTING RESULT OF THESE MEASUREMENTS IS THAT THE AVERAGE MAXIMUM COMPRESSION RATIO WITH NO DISTORTION IS LESS THAN 3:1 AND CLOSER TO ABOUT 2.5:1. IN AN ACTUAL SYSTEM, OF COURSE, THE COMPRESSION RATIO WOULD BE LOWER DUE TO THE "OVERHEAD" EFFECT OF SYNCHRONIZATION, ERROR CONTROL AND REFERENCE INFORMATION.

EVALUATION BASIS

COMPRESSION RATIO

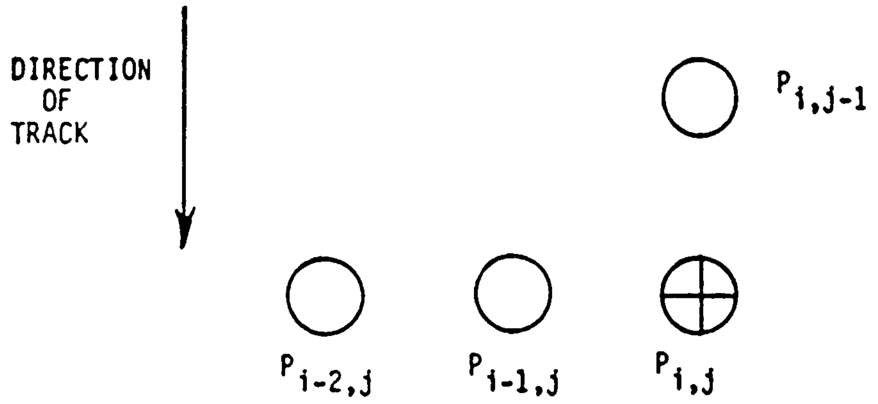
MEAN SQUARE ERROR

BUFFER REQUIREMENTS

SCHEMES CHOSEN:

DPCM FOLLOWED BY ENTROPY CODER

DPCM WITH A NON UNIFORM MAX QUANTIZER



a) NOTATION USED

b) PREDICTION SCHEMES

$$\hat{P}_{i,j} = P_{i-1,j}$$ ADJACENT

$$\hat{P}_{i,j} = 2P_{i-1,j} - P_{i-2,j}$$ LINEAR EXTRAPOLATION

$$\hat{P}_{i,j} = \frac{P_{i-1} + P_{i,j-1}}{2}$$ TWO DIMENSIONAL: TWO ADJACENT

ACHIEVABLE COMPRESSION RATIO USING DPCM FOLLOWED BY HUFFMAN-TYPE ENTROPY CODE ASSUMING LAPLACIAN P.D.F.

PREDICTOR	URBAN	GEOLOGICAL
ONE DIMENSION	2.0	2.15
TWO DIMENSION	2.16	2.33
ADAPTIVE TWO DIMENSION	2.23	2.36
CALCULATED USING CONDITIONAL ENTROPY	2.44	2.62

NON-UNIFORM QUANTISER RESULTS

For Laplacian pdf

Design Sigma	Decision Levels	Reconstruction Levels	Projected D/Sigma^2	Absolute Mean Square Error/Pixel		Normalized* Mean Squa:e Error/Pixel		Compression Ration
		3 Bit N = 7		Image 1	Image 2	Image 1	Image 2	
2.0	1,2,4	0,2,3,6	.102	4.17	3.66	.274	.335	2.7
2.5	1,2,4	0,2,3,6	.079	4.17	3.66	.274	.335	
3.0	1,3,6	0,2,5,8	.065	2.04	1.85	.134	.170	
3.5	1,3,7	0,2,5,8	.059	2.18	1.98	.143	.181	
4.0	1,4,8	0,3,6,11	.052	1.16	1.10	.076	.101	
4.5	1,4,8	0,3,6,11	.053	1.16	1.10	.076	.101	
5.0	1,4,9	0,3,6,13	.054	1.13	1.04	.074	.095	
5.5	1,5,11	0,3,8,15	.053	1.21	1.10	.079	.100	
		3 Bit N = 8						
2.0	0,1,2,4	1,2,3,6	.102	4.34	3.84	.285	.352	2.7
2.5	0,1,2,4	1,2,3,6	.073	4.03	3.50	.264	.321	
3.0	0,1,3,6	1,2,5,8	.059	1.91	1.72	.126	.157	
3.5	0,1,3,7	1,2,5,10	.053	1.23	1.11	.080	.101	
4.0	0,2,5,9	1,4,7,12	.046	.92	.89	.060	.081	
4.5	0,2,5,9	1,4,7,12	.045	.92	.89	.060	.081	
5.0	0,2,5,10	1,4,9,14	.045	1.18	1.04	.077	.095	
5.5	0,3,6,12	1,4,9,16	.044	.99	.87	.064	.081	
		4 Bit N = 15						
2.0	1,2,3,4,6,7,9	0,2,3,4,5,7,8,11	.086	.72	.72	.047	.065	2.0
2.5	1,2,3,4,6,8,11	0,2,3,4,5,7,10,13	.055	.51	.53	.033	.048	
3.0	1,2,3,4,7,9,13	0,2,3,4,6,8,11,16	.041	.45	.45	.029	.041	
3.5	1,2,3,4,7,10,14	0,2,3,4,6,9,12,17	.031	.44	.44	.028	.040	
4.0	1,2,3,5,8,11,15	0,2,3,4,7,10,13,18	.026	.44	.44	.029	.040	
4.5	1,2,3,5,8,11,15	0,2,3,4,7,10,13,18	.022	.44	.44	.029	.040	
5.0	1,2,3,5,8,11,15	0,2,3,4,7,10,13,18	.019	.44	.44	.029	.040	
5.5	1,2,3,5,8,11,15	0,2,3,4,7,10,13,19	.017	.45	.44	.029	.040	

PERFORMANCE SUMMARY

Method	Maximum Compression Ratio	Problems	Advantages
DPCM + Huffman	2-2.3	• Errors in Tx can cause loss of code sync	• With error correcting code could produce very high fidelity
Parameters:		• Requires Pre Proc. (ie Radiom. Corr.)	
2-adjacent prediction modified code adaptive by line segment		• Potential buffer overflow • Large memory needed on board	
DPCM with Nonuniform Quantiser		• Distortion D/σ^2	• Simple to implement • Degrades gracefully as image gets "busier"
N = 7/8	2.67	~0.09	
N = 15	2.0	~0.04	• Fixed block structure

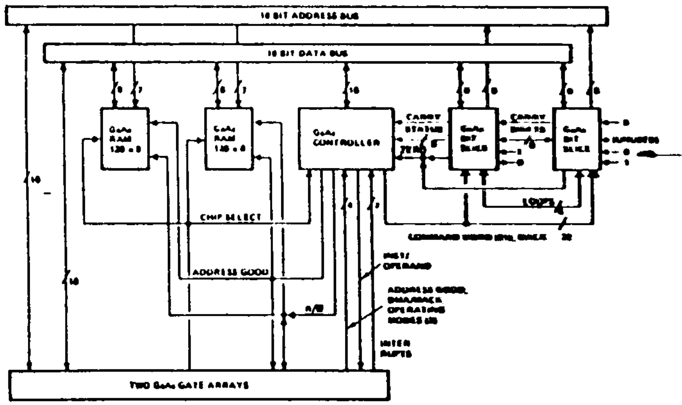
TECHNOLOGY TRADE OFF

PREDICTED SPEED-POWER CHARACTERISTIC OF HONEYWELL VHSIC CHIPS, ECL AND GaAs TECHNOLOGIES WHEN PERFORMING THE NON-UNIFORM QUANTIZING DPCM ALGORITHM (7 MICRO INSTRUCTIONS)

	DEDICATED DESIGN		PROGRAMMABLE PROCESSOR	
	ECL	GaAs	VHSIC (Si) (HONEYWELL)	APP (GaAs)
INSTRUCTION TIME (NSEC.)	-	-	40 NSEC.	5 NSEC.
PROCESSING TIME (NSEC.)	58.5	9.7	240	45
POWER, WATTS	14.0	2.7	4.7	4.5

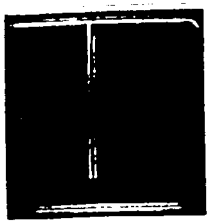
GAAS ADAPTIVE PROGRAMMABLE PROCESSOR (APP)

BLOCK DIAGRAM -GaAs 1750A COMPUTER



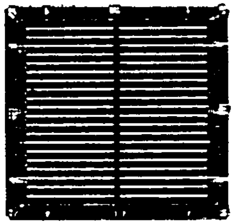
16K bit RAM

(DARPA)



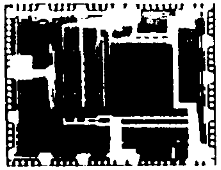
7K CONFIGURABLE GATE ARRAY

(DARPA)

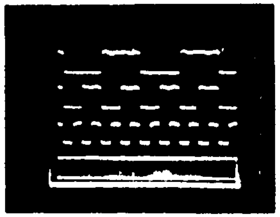


8-bit SLICE GENERAL PROCESSOR

(NASA)

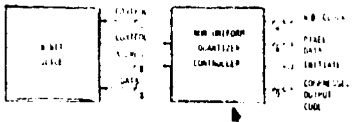


ALU Add, Incrementing
Register 0, 150 MHz Clock
Shows: bit 5
bit 4
bit 3
Clock



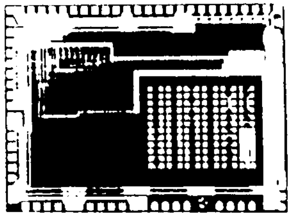
BLOCK DIAGRAM

GaAs DATA COMPRESSION PROCESSOR



NON-UNIFORM QUANTIZER CONTROLLER

(NASA)



Overview

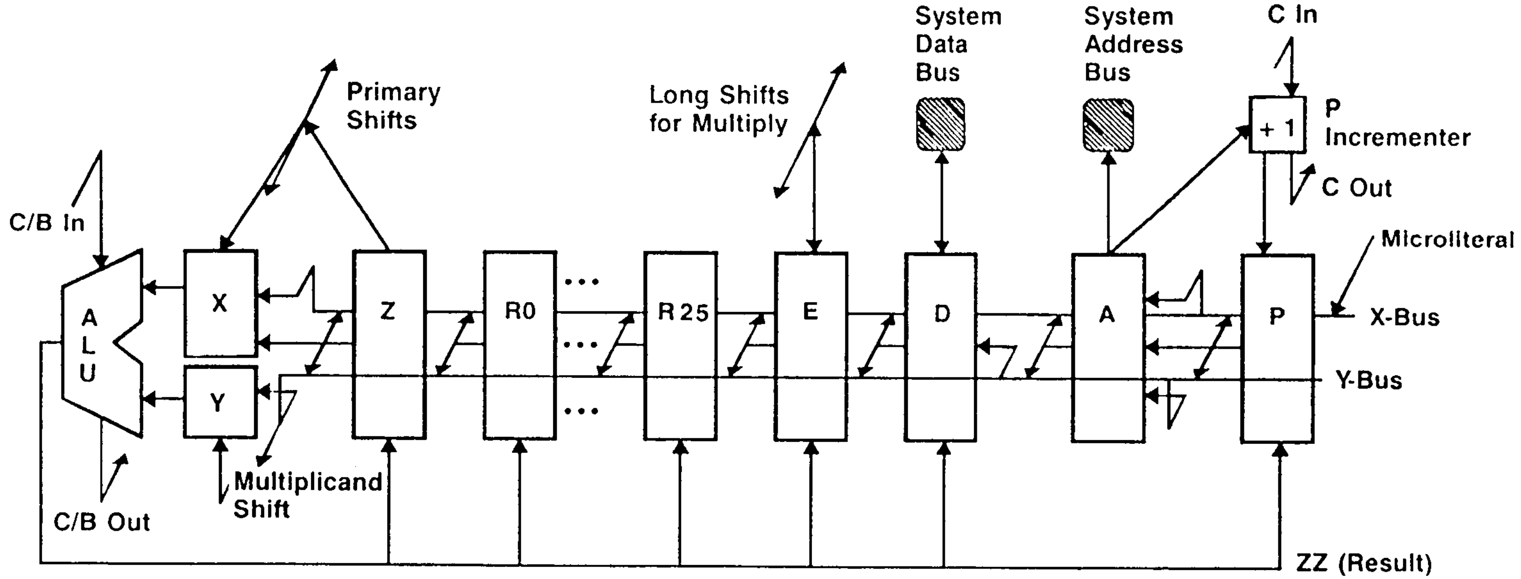
8-Bit Slice Processor
- Original Architecture
- MIL-STD-1750A Computer Building Block
- Flexible Command Structure
- Signal Processing Applications, RISC

GaAs D-MESFET Technology
- Non S/A Ion Implanted
- Threshold Voltage −1.0V
- Gate Length 1.0 μm
- Conductor Pitch 4 μm (M1), 6 μm (M2)

6.6 ns Register–Register Add/Subtract
- 150 MHz Clock

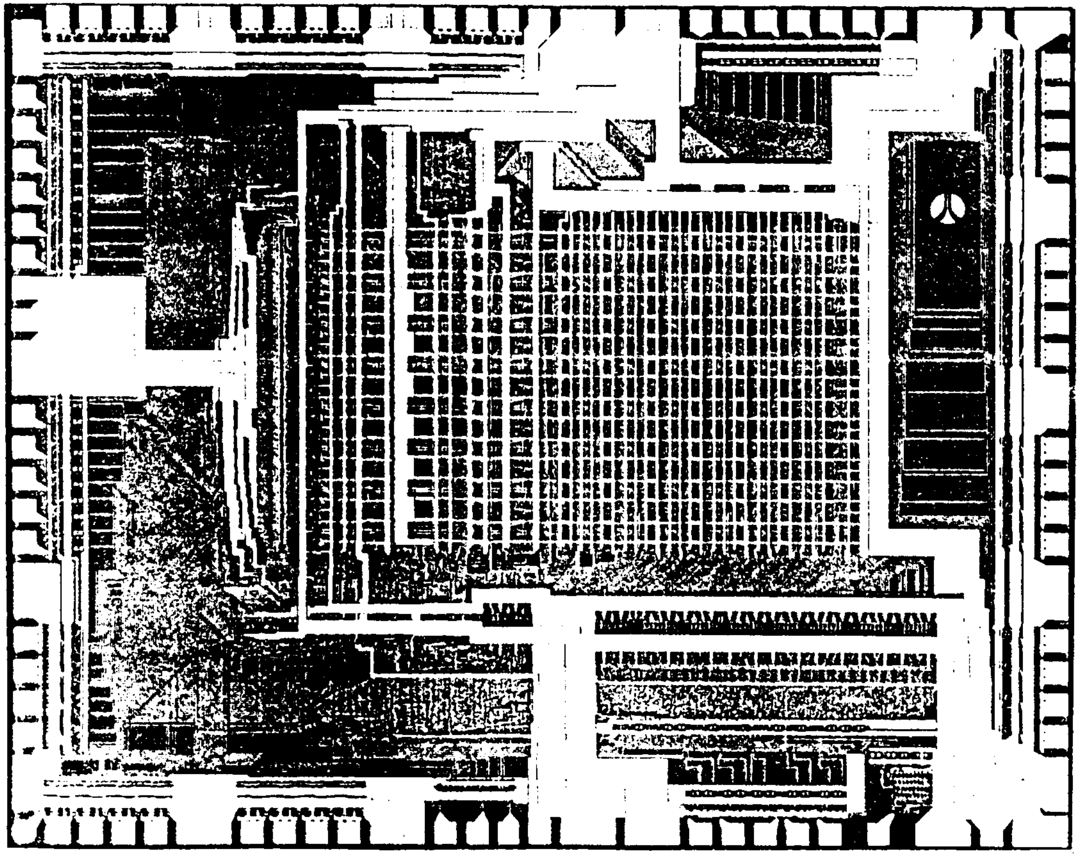
Power 4.2W (Low I_{DSS})
9.2W (High I_{DSS})

Data Path Functional Block Diagram



- 31 Word × 8-Bit 2 Port Register File
- 8 Function ALU
- 3 Internal Buses
- Program Counter Incrementer
- Shifts
- 8-Bit Address Port
- 8-Bit Bidirectional Data Port

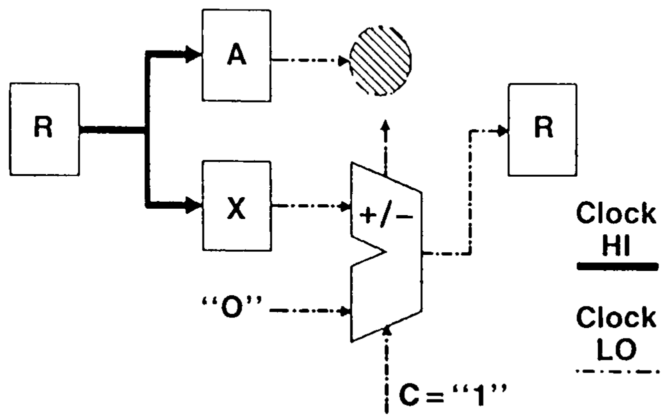
Die Photomicrograph



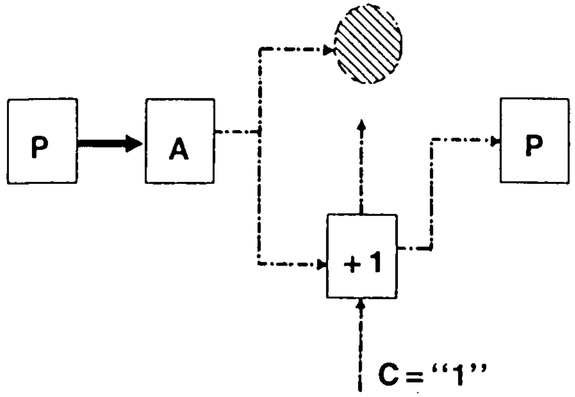
- Die Size
  4.9mm x 3.9mm
- Device Count
  9400 Transistors
  3010 Diodes
- Pads
  64 Signal
  29 Power/Ground

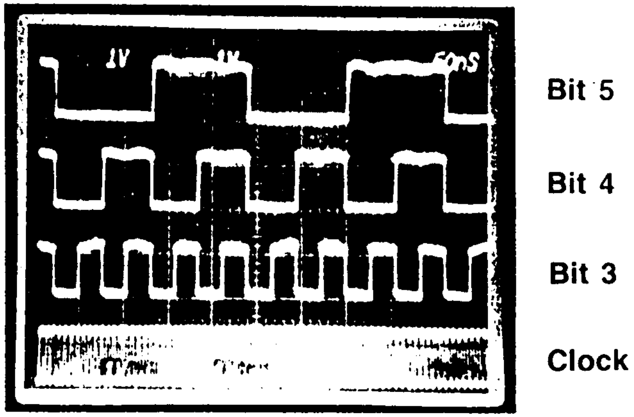
Performance Results

Register to Register Add/Subtract



Program Counter Increment





Memory Address Output Oscilloscope Waveforms

ALU Adds Incrementing Register 0, at 150 MHz Clock

- **Single Chip Worst-Case Delay Paths 6.6 ns**

☆U.S. GOVERNMENT PRINTING OFFICE: 1989 - 627064 / 86024

National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CP-3025	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Proceedings of the Scientific Data Compression Workshop		5. Report Date February 1989 6. Performing Organization Code 636.0
7. Author(s) H. K. Ramapriyan, Editor		8. Performing Organization Report No. 89B0038 10. Work Unit No. 584-02-11
9. Performing Organization Name and Address Goddard Space Flight Center Greenbelt, Maryland 20771		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001		13. Type of Report and Period Covered Conference Publication 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract Continuing advances in space and Earth science knowledge require increasing amounts of data to be gathered from spaceborne sensors. NASA expects to launch sensors during the next two decades which will be capable of producing an aggregate of 1500 Megabits per second if operated simultaneously. Primary scientific disciplines contributing such data rates are remote sensing and materials science. Such high data rates cause stresses in all aspects of end-to-end data systems. New technologies and techniques are needed to relieve such stresses. Potential solutions to the massive data rate problems are: data editing, greater transmission bandwidths, higher density and faster media, and data compression. The Data Compression Workshop, held in Snowbird, Utah, during May 3-5, 1988 considered one particular solution to this data problem. Over 75 participants from 36 different government, university and industrial organizations attended the workshop where scientists and data compression technologists were brought together to better understand science mission requirements and potential applications of data compression techniques to future missions. Through four subpanels on Science Payload Operations, Multispectral Imaging, Microwave Remote Sensing and Science Data Management, the attendees made several recommendations for research in data compression and its applications to scientific data from space platforms.

17. Key Words (Suggested by Author(s))
Data Compression Algorithms and Hardware;
On-board Processing;
Data Storage and Management;
Remote Sensing
	18. Distribution Statement
	Unclassified - Unlimited
	
	Subject Category 61

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 476	22. Price A21